technische universität dortmund

IT in Produktion und Logistik

Masterarbeit

zur Erlangung des Grades M. Sc.

Experimenteller Vergleich von Datenvorverarbeitungsverfahren zur Datenreduktion und zur Behandlung fehlender Werte im Kontext von Data Mining

Vorgelegt von:	Jakob Simon
Matrikelnummer:	220127
Studiengang:	Wirtschaftsingenieurwesen

Ausgabedatum:	03.04.2025
Abgabedatum:	18.09.2025

Erstprüfer:	Dr.-Ing. Dipl.-Inform. Anne Antonia Scheidler
Zweitprüfer:	Florian Hochkamp, M. Sc.

Technische Universität Dortmund

Fakultät Maschinenbau

Fachgebiet IT in Produktion und Logistik

Inhaltsverzeichnis

Abbildungsverzeichnis III
Tabellenverzeichnis IV
Abkürzungsverzeichnis V
1 Einleitung 1
2 Grundlagen der Datenvorverarbeitung im Data Mining 3
2.1 Grundbegriffe der Datentechnik 3
2.2 Grundlagen des Data Mining 5
2.2.1 Arten von Data-Mining-Verfahren 7
2.2.2 Evaluationsmetriken zur Bewertung von Data-Mining-Verfahren 8
2.3 Einführung in die Datenvorverarbeitung 11
3 Umgang mit fehlenden Werten 13
3.1 Typen fehlender Werte 13
3.2 Verfahren zur Erkennung fehlender Werte 13
3.3 Verfahren zur Behandlung fehlender Werte 15
3.3.1 Eliminierungsverfahren 15
3.3.2 Einführung in die Imputationsverfahren 17
3.3.3 Statistische Imputationsverfahren 18
3.3.4 Klassenbasierte Imputationsverfahren 20
3.3.5 Multivariate Imputationsverfahren 21
3.3.6 Imputation mit maschinellen Lernverfahren 24
3.4 Schlussfolgerungen zum Umgang mit fehlenden Werten 26
4 Umgang mit Datenredundanzen 27
4.1 Typen von Datenredundanzen 27
4.2 Verfahren zur Reduktion von Datenredundanzen auf Merkmalsebene 27
4.2.1 Korrelationsanalyse 28
4.2.2 Mutual-Information-Verfahren 28
4.2.3 Variance-Inflation-Factor-Verfahren 29
4.2.4 Wrapper-Verfahren 29
4.2.5 Embedded-Verfahren 29
4.2.6 Merkmalsextraktionsverfahren 30
4.3 Verfahren zur Reduktion von Datenredundanzen auf Instanzebene 30
4.3.1 Distanzbasierte Analyse 31
4.3.2 Clustering 32
4.3.3 Instanzenselektion 32
4.4 Schlussfolgerungen zum Umgang mit Datenredundanzen 32
5 Konzeption des Experimentierplans 33

5.1 Kriterien zur Auswahl geeigneter Evaluationsmetriken 34
5.1.1 Ziel und Funktion von Evaluationsmetriken 34
5.1.2 Methodische Anforderungen an Evaluationsmetriken 35
5.1.3 Ermittlung geeigneter Bewertungsdimensionen 36
5.1.4 Auswahl geeigneter Evaluationsmetriken 38
5.2 Kriterien zur Auswahl geeigneter Datenvorverarbeitungsverfahren 41
5.3 Aufbau des Experimentierplans 43
5.3.1 Konzeptionelle Zielsetzung und Bewertungsstrategie 43
5.3.2 Untersuchung von Einzelverfahren 44
5.3.3 Untersuchung von Verfahrenskombinationen und Reihenfolgevariation 45
6 Praktische Umsetzung des Experimentierplans 47
6.1 Datengrundlage und Beschreibung 47
6.2 Auswahl des Klassifikationsmodells und Baseline-Versuch 49
6.3 Auswahl der Datenvorverarbeitungsverfahren 51
6.3.1 Auswahl der Verfahren zur Behandlung fehlender Werte 52
6.3.2 Auswahl der Verfahren zur Datenreduktion 54
6.4 Durchführung der Experimente 57
6.5 Auswertung der Versuchsergebnisse 59
6.5.1 Auswertung der Einzelverfahren 60
6.5.2 Auswertung der Verfahrenskombinationen 62
7 Diskussion und Fazit 67
7.1 Zusammenfassung der Ergebnisse 67
7.2 Interpretation der Ergebnisse 67
7.3 Kritische Reflexion 69
7.4 Fazit 70
8 Zusammenfassung und Ausblick 71
Literaturverzeichnis V
Anhang XII
Anhang A: Übersicht der implementierten Versuchskonfigurationen XII
Anhang B: Übersicht der Versuchsergebnisse XIV
Anhang C: Ergebnisdiagramme Medianimputation und kNN-Imputation XVII
Anhang D: Quellcode XVIII
Anhang E: Versionsübersicht Python-Bibliotheken XXI
Eidesstattliche Versicherung XXII

Abbildungsverzeichnis

Abbildung 2.1: Teil der Wissenstreppe nach North (2021, S.37) 3
Abbildung 2.2: KDD-Prozess nach Han et al. (2012, S.7) 6
Abbildung 3.1: Missingness Matrix in Anlehnung an Little und Rubin (2019, S. 9) 14
Abbildung 3.2: Vergleich von Complete-Case-Analysis (b) und Available-Case-Analysis (c) anhand einer Missingness Matrix (a) 16
Abbildung 3.3: Anwendungsbeispiel für die Imputationsverfahren LOCF (b) und NOCB (c) am Ausgangsdatensatz (a) 20
Abbildung 3.4: Vereinfachtes Ablaufdiagramm der Multiple Imputation 22
Abbildung 3.5: Prinzip der regressionsbasierten Imputation (deterministisch) 24
Abbildung 4.1: Schematischer Vergleich der euklidischen und der Manhattan-Distanz 31
Abbildung 5.1: Schematische Darstellung der Anwendung von Einzelverfahren im Experimentierplan 44
Abbildung 5.2: Schematische Darstellung der Verfahrenskombination im Experimentierplan 46
Abbildung 6.1: Struktur und zentrale Kennzahlen des SECOM-Datensatzes (Python-Output) 48
Abbildung 6.2: Missingness-Matrix des SECOM-Datensatzes (Ausschnitt) 49
Abbildung 6.3: Ergebnisse des Baseline-Versuchs auf dem unvorbereiteten Rohdatensatz 50
Abbildung 6.4: Ergebnisse des Baseline-Versuchs mit SMOTE 51
Abbildung 6.5: Ablaufdiagramm der experimentellen Durchführung 59
Abbildung 6.6: Versuchsergebnisse der einzeln angewandten Imputationsverfahren 60
Abbildung 6.7: Versuchsergebnisse der einzeln angewandten Reduktionsverfahren 61
Abbildung 6.8: Versuchsergebnisse der Verfahrenskombinationen mit der Mittelwertimputation 62
Abbildung 6.9: Versuchsergebnisse der Verfahrenskombinationen mit der iterativen Imputation 64
Abbildung 6.10: Versuchsergebnisse der Verfahrenskombinationen mit der Konstantenimputation 65

Tabellenverzeichnis

Tabelle 3.1: Vergleich der Eliminierungsverfahren in Anlehnung an Bankhofer (1995, S. 102) 17
Tabelle 5.1: Methodische Anforderungen an Evaluationsmetriken 36
Tabelle 5.2: Auswahl der Evaluationsmetriken 41
Tabelle 5.3: Kriterien zur Auswahl von DVV 42
Tabelle 6.1: Bewertung der ausgewählten Imputationsverfahren anhand methodischer Kriterien 53
Tabelle 6.2: Bewertung der ausgewählten Datenreduktionsverfahren anhand methodischer Kriterien 56
Tabelle A1: Versuchsübersicht des Experimentierplans XII
Tabelle B1: Versuchsergebnisse nach Ausführung der Experimente XIV
Tabelle E1: Versionen der verwendeten Python-Bibliotheken XXI

Abkürzungsverzeichnis

AUC	Area Under the ROC Curve
CFS	Correlation-based Feature Selection
DBMS	Database Management System
DVV	Datenvorverarbeitungsverfahren
EDA	Explorative Datenanalyse
EM	Expectation-Maximization
ENN	Edited Nearest Neighbor
FM	F-Measure
HTML	Hypertext Markup Language
IQR	Interquartilsabstand
KDD	Knowledge Discovery from Data
kNN	k-Nearest-Neighbor
LOCF	Last Observation Carried Forward
MAR	Missing at Random
MCAR	Missing Completely at Random
MI	Mutual Information
MICE	Multiple Imputation by Chained Equations
MIDAS	Multiple Imputation with Denoising Autoencoders
MNAR	Missing Not at Random
MW	Mittelwert
NOCB	Next Observation Carried Backward
OOB	Out-of-Bag-Fehler
PCA	Principal Component Analysis
RFE	Recursive Feature Elimination
ROC	Receiver Operating Characteristic
SD	Standardabweichung
SECOM	Semi-Conductor Manufacturing
SMOTE	Synthetic Minority Over-sampling Technique
VIF	Variance Inflation Factor
XML	Extensible Markup Language

1 Einleitung

In einer zunehmend datengetriebenen Welt stellt der effektive Umgang mit Daten eine zentrale Herausforderung für Wissenschaft und Praxis dar (von Enzberg et al., 2024). Im industriellen Umfeld entstehen im Zuge der Digitalisierung und der damit verbundenen Vernetzung kontinuierlich große Mengen an Daten, deren systematische Nutzung eine wesentliche Voraussetzung für Wertschöpfung ist (von Enzberg et al., 2024). Systematisch analysiert, ermöglichen diese Daten Optimierungen, Mustererkennung und Prognosen (Han et al., 2012). Data-Mining-Verfahren können großen Datenmengen relevante Informationen extrahieren und diese in entscheidungsunterstützende Systeme integrieren (Han et al., 2012).

Allerdings ist die Qualität der Ergebnisse unmittelbar abhängig von der Qualität und Struktur der verwendeten Daten (Han et al., 2012). Gerade im industriellen Kontext sind reale Datensätze oft fehlerbehaftet oder unvollständig (Fleck et al., 2020). Sensoren liefern häufig fehlerhafte oder gar keine Werte, etwa infolge von Verbindungsproblemen oder Geräteausfällen, wodurch die Datenqualität erheblich beeinträchtigt werden kann (Teh et al., 2020). Dies führt zu fehlenden Werten und Redundanzen, die die Anwendung datenbasierter Analyseverfahren erheblich erschweren oder verfälschen können (Han et al., 2012). Daher ist eine sorgfältige Datenvorverarbeitung unerlässlich, um das volle Potenzial datengetriebener Modelle ausschöpfen zu können (Rahm & Do, 2000).

Vor diesem Hintergrund beschäftigt sich die vorliegende Arbeit mit der Frage, wie Datenvorverarbeitung im Hinblick auf fehlende Werte und Datenreduktion so gestaltet werden kann, dass eine hohe Analysequalität in Data-Mining-Verfahren erreicht wird. Eine grundlegende Herausforderung besteht darin, dass es keine universell besten Verfahren gibt, sondern deren Effektivität stark vom jeweiligen Anwendungs- und Datenkontext abhängt (Little & Rubin, 2019). Entsprechend ist es notwendig, verschiedene Verfahren nicht nur theoretisch zu analysieren, sondern auch in experimentellen Vergleichen auf einem gemeinsamen Datensatz systematisch gegenüberzustellen (García et al., 2015). Nur so lässt sich nachvollziehen, wie sich die Verfahren unter gleichen Bedingungen verhalten und welche Ansätze sich tatsächlich bewähren. Darüber hinaus wird untersucht, welchen Einfluss die Kombination und Reihenfolge von Datenvorverarbeitungsverfahren (DVV) auf die Analyseergebnisse haben kann.

Fehlende Werte können auf verschiedene Ursachen zurückgeführt werden, etwa auf Messfehler, defekte Sensoren oder fehlerhafte Datenübertragungen (Teh et al., 2020). Für ihre Behandlung stehen unterschiedliche Verfahren zur Verfügung, deren Wahl die Aussagekraft und Verzerrungsfreiheit der Analyse maßgeblich beeinflusst (Little & Rubin, 2019). Ungeeignete Verfahren zur Behandlung fehlender Werte können zu erheblichen systematischen Fehlern führen (Little & Rubin, 2019). Auch die Reduktion hochdimensionaler Daten ist von Bedeutung, um den Rechenaufwand von Data-Mining-Verfahren zu verringern und ihre Interpretierbarkeit sowie Generalisierungsfähigkeit zu erhöhen (Li et al., 2017).

Ziel dieser Arbeit ist es, ausgewählte DVV sowohl zur Behandlung fehlender Werte als auch zur Datenreduktion experimentell zu vergleichen und deren Leistungsfähigkeit zu bewerten. Es wird analysiert, welche Verfahren oder Verfahrenskombinationen sich im Data-Mining-Kontext als besonders geeignet erweisen und wie sich die Reihenfolge der Verarbeitungsschritte auf Modellqualität, Effizienz und Robustheit auswirkt. Die Analyse stützt sich auf einen realen industriellen Datensatz, anhand dessen die Verfahren angewendet und die resultierenden Ergebnisse systematisch ausgewertet werden. Hierzu wird ein systematischer Experimentierplan entwickelt, der die strukturierte Vergleichbarkeit der Verfahren hinsichtlich ihres Einflusses auf die Performance eines nachgelagerten Data-Mining-Verfahrens ermöglicht. Dies umfasst die Auswahl geeigneter Evaluationsmetriken zur Bewertung von Data-Mining-Verfahren, darunter Modellgüte, Effizienz in Bezug auf Laufzeit sowie Robustheit gegenüber variierenden Daten. Zudem wird die Wirkung unterschiedlicher Reihenfolgen und Kombinationen einzelner Verarbeitungsschritte gezielt untersucht.

Die Struktur der Arbeit orientiert sich an einem schrittweisen Vorgehen von der theoretischen Fundierung bis zur experimentellen Analyse. Zunächst werden zentrale Begriffe, Konzepte und Problemstellungen der Datenvorverarbeitung im Kontext von Data Mining erläutert. Neben einer begrifflichen und systematischen Einordnung werden insbesondere fehlende Werte und Datenredundanz als zentrale Herausforderungen identifiziert und hinsichtlich ihrer Ursachen, Auswirkungen und möglicher Behandlungsstrategien eingeordnet.

Darauf aufbauend folgt eine detaillierte Darstellung ausgewählter Verfahren zur Behandlung fehlender Werte sowie zur Reduktion von Merkmalen und Instanzen. Die Verfahren werden beschrieben, ihre praktischen Einsatzmöglichkeiten erläutert sowie jeweilige Stärken und Schwächen aufgezeigt. Der anschließende methodische Teil umfasst die Entwicklung eines Experimentierplans, die Auswahl relevanter Evaluationsmetriken und die Definition der experimentellen Rahmenbedingungen.

Auf dieser Grundlage erfolgt im empirischen Teil die Umsetzung des Experimentierplans anhand eines realen Datensatzes aus dem industriellen Umfeld, wobei ausgewählte DVV einzeln und in Kombination bei varierender Reihenfolge in reproduzierbaren Analysepipelines implementiert werden. Die resultierenden Experimente werden systematisch ausgewertet, wobei sowohl die Modellgüte als auch Effizienz und Robustheit betrachtet werden. Abschließend werden die Ergebnisse zusammengeführt, kritisch diskutiert und im Hinblick auf ihre praktische Relevanz sowie mögliche weiterführende Forschungsmöglichkeiten eingeordnet.

2 Grundlagen der Datenvorverarbeitung im Data Mining

Eine hochwertige Datenbasis ist eine zentrale Voraussetzung für die erfolgreiche Anwendung datenbasierter Analyseverfahren (Han et al., 2012). In der Literatur wird der Datenvorverarbeitung dabei eine entscheidende Rolle zugeschrieben, da viele Data-Mining-Verfahren stark von Qualität, Vollständigkeit und Struktur der Eingangsdaten abhängen (García et al., 2015). Insbesondere in industriellen Anwendungen sind Rohdaten häufig durch Fehler, Ausfälle oder Inkonsistenzen belastet (Teh et al., 2020). Da unzureichend aufbereitete Daten zu irreführenden Ergebnissen führen können, ist eine systematische Datenaufbereitung erforderlich, um verlässliche und akkurate Analyseergebnisse zu erhalten (García et al., 2015).

In diesem Kapitel werden grundlegende Konzepte vorgestellt, die den Rahmen für die Datenvorverarbeitung im Data-Mining-Kontext bilden. Dazu gehören zentrale Begriffe der Datentechnik, verschiedene Formen der Datenstrukturierung sowie der Aufbau von Datensätzen und Datenbanken. Darüber hinaus werden die Grundlagen des Data Mining, typische Verfahren und wichtige Evaluationsmetriken erläutert, um eine gemeinsame Basis für die anschließende Betrachtung verschiedener Arten von DVV zu schaffen.

2.1 Grundbegriffe der Datentechnik

Bevor konkrete Verfahren zur Datenvorverarbeitung näher betrachtet werden, müssen grundlegende, in dieser Arbeit verwendete, Begriffe der Datentechnik geklärt werden. Im folgenden Abschnitt erfolgt die Erläuterung zentrale Begriffe im Kontext von Strukturierung, Speicherung und Repräsentation von Daten erläutert. Dies schafft eine einheitliche begriffliche Grundlage und erleichtert den Zugang zu den weiterführenden Themen in der Datenvorverarbeitung und im Data Mining.

Datenbegriff

Zur grundlegenden Einordnung des Datenbegriffs kann auf die Wissenstreppe nach North (2021) Bezug genommen werden. Der Zusammenhang von Zeichen, Daten, Informationen und Wissen ist nach North (2021) in Abbildung 2.1 dargestellt. Sie zeigt eine stufenartige Anordnung, die den Weg von einfachen Zeichen zu komplexeren Wissensstrukturen abbildet.

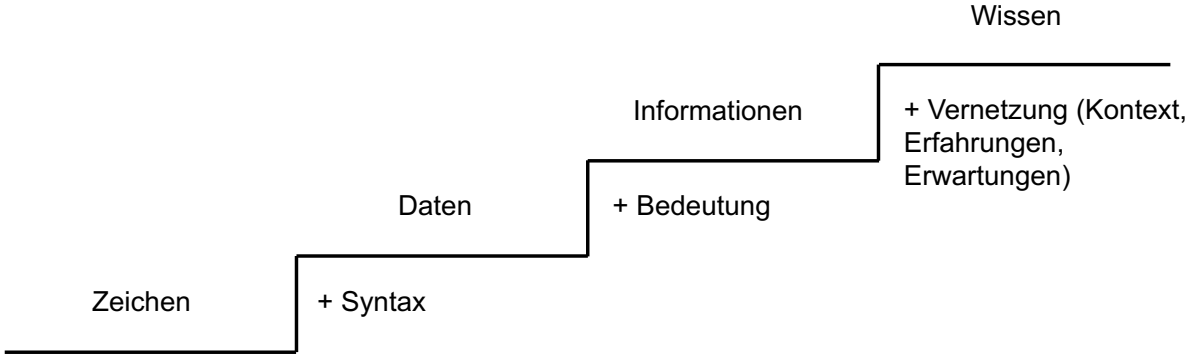


Abbildung 2.1: Teil der Wissenstreppe nach North (2021, S.37)

Abbildung 2.1 zeigt die Wissenstreppe nach North (2021), eine hierarchische Struktur, die von Zeichen über Daten und Informationen bis hin zu Wissen führt. Zeichen stellen demnach die unterste Ebene dar und umfassen einzelne Buchstaben, Ziffern oder Sonderzeichen. Erst durch syntaktische Strukturierung entstehen laut North (2021) daraus Daten. Werden Daten durch Kontext und Bedeutung angereichert, resultieren daraus Informationen (North, 2021). Wissen entsteht schließlich durch die zweckgerichtete Verknüpfung von Informationen, unter Einbeziehung von Erfahrung und Kontext (North, 2021). Aufbauend auf dieser begrifflichen Einordnung wird im Folgenden die mögliche Strukturierung von Daten betrachtet.

Datenstruktur

Im Anschluss an die Definition des Datenbegriffs wird deren mögliche Strukturierung thematisiert. Die Struktur von Daten bestimmt maßgeblich, wie diese gespeichert, verarbeitet und analysiert werden können (Krcmar, 2015). Daten können hinsichtlich ihres Strukturierungsgrades in strukturierte, semi-strukturierte und unstrukturierte Formen eingeteilt werden (Elmasri & Navathe, 2016). Strukturierte Daten folgen einem festen Schema und liegen meist in tabellarischer Form vor, wodurch sie systematisch gespeichert und effizient verarbeitet werden können (Elmasri & Navathe, 2016). Semi-strukturierte Daten, wie bspw. das auf Tags basierende Austauschformat Extensible Markup Language (XML), enthalten strukturierende Elemente, benötigen jedoch kein einheitliches Schema, da Strukturinformationen direkt mit den Daten verknüpft sind (Elmasri & Navathe, 2016). Unstrukturierte Daten hingegen, etwa in Form von Fließtexten oder Dokumenten im Hypertext Markup Language Format (HTML), enthalten kaum maschinenlesbare Strukturelemente und folgen keiner festen Organisation (Elmasri & Navathe, 2016). Die Einordnung verdeutlicht die Relevanz der Datenstruktur für die Auswahl geeigneter Analyse- und Verarbeitungsverfahren. Aufbauend auf den beschriebenen Strukturierungsformen wird im Folgenden betrachtet, wie Daten praktisch in Datensätzen organisiert und durch Datenobjekte repräsentiert werden.

Datensätze und Datenobjekte

In datenanalytischen Anwendungen werden Informationen meist in Form strukturierter Datensätze organisiert (Han et al., 2012). Jeder Datensatz besteht aus einer Menge von Datenobjekten (Instanzen), die spezifische Entitäten innerhalb eines Kontextes repräsentieren (Han et al., 2012). Objekte werden durch Attribute beschrieben, die auch als Merkmale oder Variablen bezeichnet werden (Han et al., 2012). In strukturierten Datensätzen entspricht jedes dieser Objekte einer Zeile, während die Attribute den Spalten zugeordnet sind (Han et al., 2012).

Wie bereits erwähnt, werden strukturierte Datensätze typischerweise in Tabellenform dargestellt. Liegen ausschließlich numerische Attribute vor, können sie als sogenannte Datenmatrix interpretiert werden (Tan et al., 2019). Eine Datenmatrix ist eine mathematische Struktur, in der Zeilen einzelnen Datenobjekten und Spalten den jeweiligen Attributen entsprechen (Han et al., 2012). In dieser Form lassen sich mathematische Operationen anwenden, was eine Voraussetzung für viele statistische Auswertungen und Verfahren des maschinellen Lernens darstellt (Tan et al., 2019). Sind in Datensätzen ebenfalls nichtnumerische Attribute enthalten, sind diese zunächst durch geeignete Transformation in numerische Form zu überführen (Tan et al., 2019).

Die Kenntnis der Attributstypen ist dabei von besonderer Bedeutung, da sie bestimmt, welche mathematischen und statistischen Operationen zulässig und sinnvoll sind (Tan et al., 2019). Attribute können nach ihrem Skalenniveau klassifiziert werden: nominal, ordinal oder numerisch (Han et al., 2012). Nominale Attribute kennzeichnen Kategorien ohne Rangordnung, während ordinale Attribute eine Reihenfolge besitzen (Han et al., 2012). Numerische Attribute hingegen erlauben die Messung und Berechnung von Differenzen (Han et al., 2012). Innerhalb der numerischen Attribute wird zusätzlich zwischen diskret und kontinuierlich unterschieden, abhängig davon, ob sie abzählbare (diskrete) oder stetige (kontinuierliche) Werte annehmen (Han et al., 2012). An die Darstellung von Datensätzen und Objekten schließt sich die Betrachtung von Datenbanken an.

Datenbanken

Abschließend ist zu klären, wie strukturierte Daten verwaltet und zugänglich gemacht werden können. Diese Aufgabe übernehmen Datenbanksysteme, also spezialisierte Softwaresysteme, die auf die strukturierte Speicherung, Verwaltung und den Zugriff auf große Datenmengen ausgelegt sind (Schicker, 2017). Schicker (2017) definiert eine Datenbank als „eine Sammlung von Daten, die untereinander in einer logischen Beziehung stehen und von einem eigenen Datenbankverwaltungssystem (Database Management System, DBMS) verwaltet werden“ (Schicker, 2017, S.3). Das DBMS stellt dabei eine Schnittstelle dar, über die Nutzer

auf die Daten zugreifen und diese bearbeiten können, ohne sich mit der physischer Speicherung auseinandersetzen müssen (Schicker, 2017).

Das bekannteste Modell ist die relationale Datenbank, welche Daten nach Schicker (2017) in Tabellen, sogenannten Relationen organisiert. Diese Tabellen lassen sich über Schlüsselattribute logisch miteinander verknüpfen (Schicker, 2017). Ein wesentliches Merkmal relationaler Datenbanksysteme ist die Verwendung von Structured Query Language (SQL), einer deklarativen Sprache zur Abfrage und Manipulation von Daten (Schicker, 2017). SQL ermöglicht neben der Strukturierung von Daten auch deren gezielte Abfrage, Veränderung und Verwaltung (Schicker, 2017). Dabei ist sie unabhängig von der physischen Speicherstruktur und erleichtert auf diese Weise den Zugriff auf komplexe Datenbestände (Schicker, 2017). Weiterhin charakterisieren sich relationale Datenbanksysteme über ihre Fähigkeit, auch bei parallelen Zugriffen und Störungen konsistente Datenzustände zu gewährleisten (Schicker, 2017). Dies wird durch die sogenannten ACID-Eigenschaften sichergestellt, die garantieren, dass Transaktionen entweder vollständig oder gar nicht ausgeführt werden (Atomarity), Daten stets in einem gültigen Zustand bleiben (Consistency), sich parallele Transaktionen nicht gegenseitig beeinflussen (Isolation) und einmal bestätigte Änderungen dauerhaft gespeichert werden (Durability) (Schicker, 2017).

Neben relationalen Systemen gewinnen sogenannte NoSQL-Datenbanken zunehmend an Bedeutung, da sie eine flexiblere Speicherung semi-strukturierter und unstrukturierter Daten ermöglichen und insbesondere durch schemalose oder dokumentenbasierte Ansätze dynamisch erweiterbare Datenmodelle unterstützen (Cattell, 2010). Nachdem zentrale Konzepte und Begrifflichkeiten zur Strukturierung, Speicherung und Repräsentation von Daten dargestellt wurden, richtet sich der Fokus nun auf die Datenvorverarbeitung, die nach Han et al. (2012) im Rahmen des Data Mining eine entscheidende Rolle einnimmt.

2.2 Grundlagen des Data Mining

Data Mining beschreibt nach Han et al. (2012) den Prozess, durch den aus großen Datenmengen verborgenes und bislang unbekanntes, potenziell nützliches Wissen extrahiert wird. Weiterhin beschreiben die Autoren die zentrale Zielsetzung des Data Mining als die Ableitung von verwertbaren Informationen und Wissen aus vorhandenen Rohdaten, um Entscheidungsprozesse gezielt zu unterstützen. Dabei umfasse der Begriff Data Mining nach gängiger Auffassung insbesondere jene Phase innerhalb des Prozesses Knowledge Discovery in Databases (KDD), welche sich explizit mit der automatisierten Analyse und Mustererkennung beschäftigt. Han et al. (2012) beschreiben den KDD-Prozess als eine iterative Abfolge der folgenden Schritte:

1. Data Cleaning: Bereinigung von Rauschen und inkonsistenten Daten
2. Data Integration Zusammenführen mehrerer Datenquellen
3. Data Selection: Auswahl relevanter Daten für die Analyse
4. Data Transformation: Umwandlung der Daten in für das Mining geeignete Formate
5. Data Mining: Anwendung intelligenter Verfahren zur Extraktion von Mustern
6. Pattern Evaluation: Bewertung gefundener Muster mittels Interessantheitskriterien
7. Knowledge Presentation: Präsentation des gewonnenen Wissens

Diese Schritte verdeutlichen, dass Data Mining nicht isoliert betrachtet werden kann, sondern Teil eines umfassenden Prozesses der Wissensentdeckung ist. Der beschriebene Prozess zur Wissensentdeckung (KDD-Prozess) ist in der folgenden Abbildung 2.2 dargestellt.

Abbildung 2.2 bildet den KDD-Prozess in seinen zentralen Phasen ab. Ausgangspunkt bilden die in Datenbanken vorliegenden Rohdaten, die zunächst bereinigt werden (Han et al., 2012). Im nächsten Schritt erfolgt die Auswahl und Transformation der relevanten Daten, sodass diese in einer geeigneten Form für die anschließende Analyse vorliegen (Han et al., 2012). Darauf aufbauend folgt der eigentliche Data-Mining-Schritt, in dem mit spezifischen Verfahren

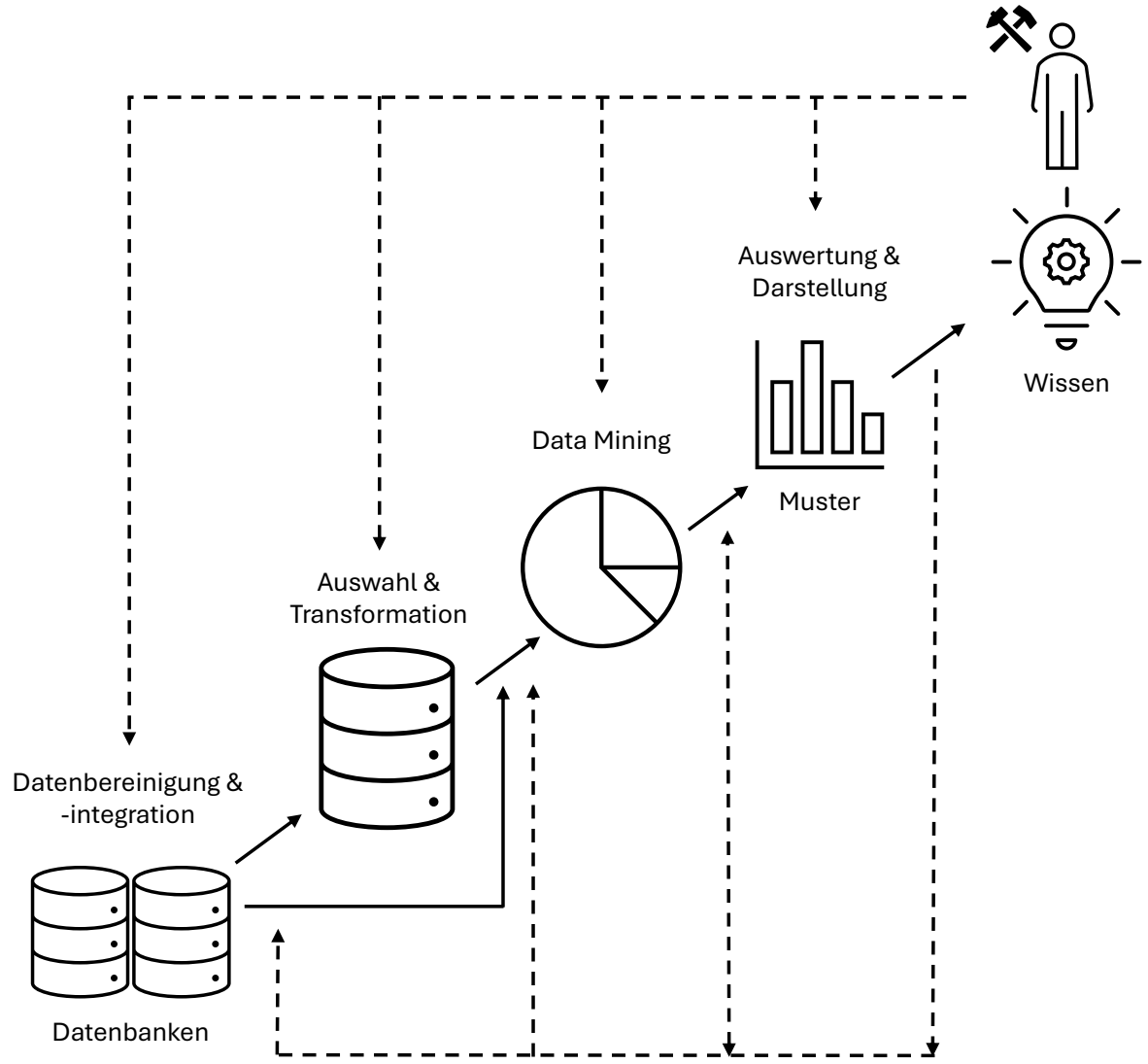


Abbildung 2.2: KDD-Prozess nach Han et al. (2012, S.7)

Muster und Strukturen in den Daten identifiziert werden (Han et al., 2012). Diese Muster werden in der nachgelagerten Phase der Auswertung und Darstellung interpretiert und in den jeweiligen Anwendungskontext eingeordnet (Han et al., 2012). Der Prozess mündet schließlich in der Generierung von Wissen (Han et al., 2012), welches die Entscheidungsgrundlage für weitere Schritte bildet. Die Rückkopplungen zwischen den einzelnen Phasen verdeutlichen, dass es sich um einen iterativen Prozess handelt, bei dem Zwischenergebnisse zu einer erneuten Anpassung früherer Schritte führen können (Han et al., 2012). Die Figur des Miners symbolisiert den Benutzer, der in jeder Phase des Prozesses eingebunden ist und durch Interaktion sowie Bewertung maßgeblich zur Generierung von Wissen beiträgt (Han et al., 2012).

Grundlegende Ziele des Data Mining sind die Klassifikation, bei der Datenobjekte vordefinierten Kategorien zugeordnet werden, die Clusteranalyse zur Gruppierung ähnlicher Objekte ohne vorgegebene Klassifizierung sowie Prognoseverfahren, die auf Basis historischer Daten zukünftige Ereignisse oder Entwicklungen vorhersagen (García et al., 2015). Die Anwendungsmöglichkeiten von Data Mining sind vielfältig und zeigen sich insbesondere im industriellen Umfeld, wo Datenanalysen zur Optimierung von Prozessen und Systemen beitragen (Fleck et al., 2020). Insgesamt zeigt sich, dass Data Mining ohne eine fundierte Datenvorverarbeitung weder effiziente noch valide Ergebnisse liefern kann (Han et al., 2012). Vor diesem Hintergrund stellt sich die Frage, mit welchen konkreten Verfahren die beschriebenen Ziele in der Praxis umgesetzt werden können. Um die Vielfalt der Ansätze einzuordnen, wird im Folgenden ein Überblick über typische Arten von Data-Mining-Verfahren gegeben.

2.2.1 Arten von Data-Mining-Verfahren

Nachdem im vorangegangenen Abschnitt die Grundlagen des Data Mining erläutert wurden, folgt nun ein Überblick über typische Verfahren, die in diesem Kontext zur Anwendung kommen. Ein zentraler Bestandteil vieler Data-Mining-Verfahren ist das maschinelle Lernen, dessen Verfahren sich in überwachte und unüberwachte Verfahren einteilen lassen (Caparrós, 2025). Beim überwachten Lernen liegt nach Caparrós (2025) eine Zielvariable vor, die anhand von Eingabedaten vorhergesagt werden soll. Das Modell lernt demnach aus vorhandenen Beispielen einen Zusammenhang zwischen Merkmalen und Zielwerten. Unüberwachtes Lernen hingegen arbeitet mit unbeschrifteten Daten und dient der Erkennung von Mustern, Strukturen oder Gruppen innerhalb der Daten (Caparrós, 2025).

Eine der verbreitetsten Anwendungsformen des maschinellen Lernens ist nach Lanquillon (2019) die Klassifikation. Klassifikation bezeichnet demnach die Zuordnung von Objekten zu einer oder mehreren vordefinierten Klassen anhand ihrer Merkmale und ist dem überwachten Lernen zugeordnet, da das Modell anhand von Beispielen mit bekannten Klassenzugehörigkeiten trainiert wird. Der Autor unterscheidet zudem zwischen binären und mehrklassigen Klassifikationsproblemen. Bei binären Klassifikationen stehen zwei Klassen zur Auswahl, etwa Spam und Nicht-Spam bei E-Mails (Lanquillon, 2019). Mehrklassenprobleme hingegen umfassen nach Lanquillon (2019) mehrere mögliche Ausgänge. Die Leistungsfähigkeit eines Klassifikators wird anhand von Gütemaßen wie Genauigkeit, Recall oder Precision bewertet, wobei die Wahl des geeigneten Maßes vom jeweiligen Anwendungsfall abhängt (Lanquillon, 2019). Um die Vorhersagequalität realistisch beurteilen zu können, erfolgt die Evaluation in der Regel auf zuvor getrennten Testdaten (Lanquillon, 2019). In der Praxis werden für die Klassifikation verschiedene Modelltypen eingesetzt. So strukturieren Entscheidungsbäume Daten durch wiederholte Aufteilung und ermöglichen verständliche Modelle für Klassifikation und Regression (Lanquillon, 2019), während k-Nearest-Neighbors (kNN) die Klassenzugehörigkeit bestimmen, indem sie eine Beobachtung der Klasse der ähnlichsten, bereits klassifizierten Beispiele zuordnen (Zhang, 2016). Support Vector Machines trennen Klassen, indem sie eine Grenze wählen, die möglichst weit von den nächstliegenden Punkten beider Klassen entfernt ist (Zeigermann & Nguyen, 2024). Der Naive-Bayes-Klassifikator hingegen berechnet laut Bishop (2006) die Wahrscheinlichkeit einer Klassenzugehörigkeit basierend auf den beobachteten Merkmalen unter der Annahme, dass diese Merkmale bedingt voneinander unabhängig sind. Trotz dieser vereinfachenden Annahme erzielt das Verfahren in vielen praktischen Anwendungen robuste Ergebnisse (Bishop, 2006).

Neben einzelnen Modelltypen kommen auch sogenannte Ensemble-Verfahren zum Einsatz, welche mehrere Modelle zu einem Gesamtmodell mit verbesserter Vorhersageleistung kombinieren (Gomes et al., 2017). Ein häufig genutztes Ensemble-Verfahren ist der Random Forest nach Breiman (2001). Dabei handelt es sich um ein Ensemble aus vielen Entscheidungsbäumen, die jeweils mit zufälligen Teilmengen der Trainingsdaten und Merkmale erzeugt werden (Breiman, 2001). Die Vorhersagen der einzelnen Entscheidungsbäume werden demnach zu einer Gesamtaussage zusammengefasst, wobei in Klassifikationsaufgaben in der Regel diejenige Klasse gewählt wird, die von den meisten Bäumen vorhergesagt wurde. Random Forests zeichnen sich durch eine hohe Robustheit gegenüber Überanpassung sowie durch gute Skalierbarkeit aus (Breiman, 2001). Der Autor schreibt ebenfalls über die Fähigkeit von Random Forests, die relative Bedeutung einzelner Merkmale im Modell zu bewerten. Das von Friedman (2001) entwickelte Gradient Boosting wiederum beschreibt ein Verfahren, bei dem Entscheidungsbäume sequenziell trainiert werden. Jeder neu hinzugefügte Baum modelliert dabei die Residuen, also die Abweichungen zwischen Vorhersagen und tatsächlichen Zielwerten, der vorherigen Bäume, sodass die Fehler schrittweise reduziert und die Vorhersagen fortlaufend verbessert werden (Friedman, 2001). Eine Weiterentwicklung dieses Ansatzes stellt das Histogramm-basierte Gradient Boosting dar (Ke et al., 2017). Dabei werden die Wertebereiche der Merkmale in Intervalle (Bins) eingeteilt, sodass die Suche nach Trennpunkten nicht mehr über jeden einzelnen Wert erfolgen muss, sondern nur über die Grenzen dieser Intervalle (Ke et al., 2017). Dadurch wird das Training erheblich beschleunigt und kann auch bei sehr großen Datensätzen effizient durchgeführt werden (Ke et al., 2017).

Unüberwachte Lernverfahren kommen zur Anwendung, wenn keine Zielvariable vorliegt (Caparrós, 2025). Ziel ist es laut Caparrós (2025), innere Strukturen in Daten zu erkennen oder ähnliche Objekte zu gruppieren. Als Beispiel dafür nennt der Autor die Clusteranalyse, bei der etwa mithilfe des k-Means-Clustering Datenpunkte auf Grundlage ihrer Merkmalsähnlichkeit in Gruppen aufgeteilt werden. Weitere Verfahren, wie die Principal Component Analysis (PCA), dienen der explorativen Analyse großer Datensätze, indem sie die in vielen korrelierten Variablen enthaltene Information auf wenige unkorrelierte Hauptkomponenten verdichten (Saporta & Niang, 2009). Ziel der PCA ist es, durch lineare Kombinationen der ursprünglichen Variablen eine möglichst verlustarme und verständliche Darstellung der Datenstruktur in einem Raum mit reduzierter Dimension zu ermöglichen (Saporta & Niang, 2009).

Verfahren des maschinellen Lernens sind in vielen Data-Mining-Verfahren ein grundlegendes Werkzeug zur Mustererkennung und Entscheidungsunterstützung (Witten et al., 2017). Die Wahl eines geeigneten Verfahrens hängt stets von den Eigenschaften der Daten sowie den vorhandenen Informationen ab (Witten et al., 2017). Um die Leistungsfähigkeit der eingesetzten Verfahren fundiert bewerten zu können, ist der Einsatz geeigneter Evaluationsmetriken unerlässlich (Witten et al., 2017). Sie ermöglichen eine objektive Beurteilung der Modellgüte und bilden somit die Grundlage für den Vergleich und die Auswahl unterschiedlicher Verfahren (Witten et al., 2017).

2.2.2 Evaluationsmetriken zur Bewertung von Data-Mining-Verfahren

In der Bewertung von Data-Mining-Verfahren spielen Evaluationsmetriken eine wichtige Rolle, da sie bestimmen, wie die Qualität und Nützlichkeit eines Modells eingeschätzt wird (Witten et al., 2017). Ohne geeignete Metriken ist es kaum möglich, fundierte Aussagen über die Leistungsfähigkeit etwa eines Klassifikators zu treffen oder zwischen mehreren Modellen zu entscheiden, insbesondere im Hinblick auf deren Generalisierungsfähigkeit auf unbekannte Daten (Witten et al., 2017). Die Wahl der Evaluationsmetriken beeinflusst laut Hossin und Sulaiman (2015) nicht nur die abschließende Beurteilung eines Modells, sondern hat bereits während der Modellentwicklung eine grundlegende Funktion. So wirkt sie sich bspw. auf den Vergleich unterschiedlicher Modelle während des Trainings aus und beeinflusst Optimierungsprozesse, etwa bei der Anwendung von Heuristiken oder evolutionären Algorithmen (Hossin & Sulaiman, 2015). Eine der am häufigsten verwendeten Metriken zur Bewertung der Güte von Klassifikationsmodellen ist die Accuracy (Hossin & Sulaiman, 2015). Sie misst die Fähigkeit eines Modells, neue und ungesehene Daten korrekt vorherzusagen (Hossin & Sulaiman, 2015). Berechnet wird die Accuracy als der Anteil korrekt klassifizierter Instanzen an der Gesamtzahl der getesteten Instanzen und bietet somit eine einfache und intuitive Maßzahl zur Gesamtleitung eines Modells (Hossin & Sulaiman, 2015). Die folgende Formel zeigt die mathematische Berechnung der Accuracy (Hossin & Sulaiman, 2015).

$$Accuracy = \frac{tp + tn}{tp + fp + tn + fn}$$

Dabei steht tp für True Positive, tn für True Negative, fp für False Positive und fn für False Negative. Wenn die Klassenverteilung im zu entsprechenden Datensatz ausgewogen ist, kann Accuracy laut Witten et al. (2017) eine sinnvolle erste Einschätzung der Modellleistung geben. Allerdings warnen die Autoren gleichzeitig, dass Accuracy bei stark unausgewogenen Klassenverhältnissen irreführend sein kann, da sie in diesem Fall trotz schlechter Modelleistung hohe Werte liefern kann. So kann ein Modell, welches ausschließlich die Mehrheitsklasse vorhersagt, zwar eine hohe Accuracy erzielen, ist jedoch praktisch nutzlos, weil es die Minderheitsklasse vollständig ignoriert (Bekkar et al., 2013). Neben der Accuracy wird häufig auch die Fehlerrate betrachtet, also der Anteil der falsch klassifizierten Instanzen, der insbesondere bei sensiblen Anwendungen mit hohen Fehlkosten entscheidungsrelevant sein kann (Hossin & Sulaiman, 2015). Trotz ihrer Einfachheit eignet sich die Fehlerrate jedoch nur bedingt zur differenzierten Bewertung, da sie keine Aussage über die Art der Fehler oder die betroffenen Klassen trifft (Hossin & Sulaiman, 2015). Andere Metriken zur Bewertung der Güte wie Precision, Recall und die daraus abgeleitete F-Measure (FM) adressieren die genannten Schwächen der Accuracy, indem sie die Leistungsfähigkeit eines Modells differenzierter untersuchen

(Hossin & Sulaiman, 2015). Precision gibt an, welcher Anteil der als positiv klassifizierten Instanzen tatsächlich positiv ist, und bewertet damit die Genauigkeit der positiven Vorhersagen eines Modells (Hossin & Sulaiman, 2015). Anhand folgender Formel lässt sich die Precision berechnen (Hossin & Sulaiman, 2015).

$$Precision\ (p) = \frac{tp}{tp + fp}$$

Recall hingegen quantifiziert den Anteil der tatsächlich positiven Fälle, die vom Modell korrekt erkannt wurden (Manning et al., 2008). Da die Summe aus *tp* und *fn* die Menge aller Positiven abbildet, wird Recall anhand der folgenden Formel berechnet (Manning et al., 2008).

$$Recall\ (r) = \frac{tp}{tp + fn}$$

Die FM kombiniert diese beiden Größen zu einem harmonischen Mittelwert und wird häufig verwendet, wenn ein Gleichgewicht zwischen Präzision und Vollständigkeit gefordert ist (Hossin & Sulaiman, 2015). Die folgende Formel zeigt die mathematische Berechnung der FM (Hossin & Sulaiman, 2015).

$$FM = \frac{2 * p * r}{p + r}$$

Diese Metriken gelten als aussagekräftigere Alternativen zur Accuracy, insbesondere bei unausgewogenen Klassifikationsproblemen (Hossin & Sulaiman, 2015). Während Accuracy, Precision, Recall und FM die Modellgüte im Hinblick auf einzelne Klassen und Fehlerarten bewerten, zielen andere Metriken darauf ab, die generelle Trennschärfe eines Klassifikators zu bewerten (Hossin & Sulaiman, 2015). Eine zentrale Metrik dafür ist die Area Under the ROC Curve (AUC) (Hossin & Sulaiman, 2015). Die AUC basiert auf der Receiver Operating Characteristic (ROC)-Kurve (Hossin & Sulaiman, 2015), welche die True Positive Rate gegen die False Positive Rate aufträgt und damit die Fähigkeit eines Klassifikators visualisiert, korrekte von falschen positiven Vorhersagen über unterschiedliche Schwellen hinweg zu unterscheiden (Fawcett, 2006). Ein Nachteil der AUC besteht in ihrem hohen Rechenaufwand, insbesondere bei Multiklassenproblemen, wodurch ihre praktische Anwendbarkeit in großen Lösungsräumen eingeschränkt sein kann (Hossin & Sulaiman, 2015).

Für eine bessere Bewertung der Klassifikationsleistung bei unausgeglichenen Datensätzen empfehlen Hossin und Sulaiman (2015) hybride Evaluationsmetriken. Die Optimized Precision ist eine erweiterte Bewertungsmetrik, die Precision mit einem Gewichtungsfaktor aus Sensitivität und Spezifität kombiniert, um eine ausgewogenere Leistungsbewertung bei unausgeglichenen Datensätzen zu ermöglichen (Ranawana & Palade, 2006). Weiterentwicklungen wie die Optimized Accuracy with Extended Recall-Precision erweitern diesen Ansatz, indem sie gewichtete Zusammenhänge zwischen Precision und Recall berücksichtigen, was eine feinere Bewertung bei komplexen oder widersprüchlichen Klassifikationssituationen ermöglicht (Hossin & Sulaiman, 2015). Diese Verfahren wurden bislang vorrangig im Kontext binärer Klassifikation untersucht, ihre Eignung für Multiklassenszenarien ist jedoch nicht abschließend geklärt (Hossin & Sulaiman, 2015).

In der Praxis ist neben Evaluationsmetriken zur Bewertung der Modellgüte insbesondere auch die Effizienz eines Modells von entscheidender Bedeutung (Schwartz et al., 2020). Sie bezieht sich im Kontext von Data-Mining-Verfahren sowohl auf die Zeit, die zum Trainieren des Modells benötigt wird, als auch auf die Dauer der Vorhersage (Inferenzzeit) bei neuen Daten (Schwartz et al., 2020). Gerade bei großen Datensätzen oder in Anwendungen mit Echtzeitanforderungen kann ein rechenintensives Modell ungeeignet sein, selbst wenn es eine hohe Vorhersagegenauigkeit bietet. Wie Bishop (2006) und Schwartz et al. (2020) betonen, können insbesondere komplexe Modelle wie neuronale Netze erheblichen Rechen- bzw. Ressourcenaufwand verursachen. Die Effizienz ist daher ein wichtiges Kriterium bei der Modell- und Verfahrensauswahl und sollte im Sinne der Gesamtbewertung stets gemeinsam mit Gütemaßen betrachtet werden (Strubell et al., 2019). In der Praxis lässt sich die Effizienz von Machine-

Learning-Verfahren anhand der Trainingszeit erfassen, die von vielen Programmbibliotheken wie etwa `scikit-learn` systematisch vergleichbar gemacht wird (Pedregosa et al., 2011).

Eine weitere wichtige Dimension in der Bewertung von Machine-Learning-Modellen ist die Robustheit (Braiek & Khomh, 2024). Die Autoren definieren die Robustheit von Machine-Learning-Modellen auf Basis von Freiesleben und Grote (2023) als „capacity of a model to sustain stable predictive in the face of variations and changes in the input“ (Braiek & Khomh, 2024, S. 2). Während ein Modell auf Trainings- und Testdaten eine hohe Genauigkeit erreichen kann, zeigt sich seine Robustheit demnach erst, wenn es mit Eingaben konfrontiert wird, die von der Trainingsverteilung abweichen, etwa durch Rauschen, Auslassungen oder gezielte Manipulationen (Freiesleben & Grote, 2023). Vor allem im sicherheitskritischen oder industriellen Umfeld ist Robustheit essenziell, da hier kleine Fehler große Auswirkungen haben können (Braiek & Khomh, 2024). Bishop (2006) weist darauf hin, dass übermäßig komplexe Modelle anfälliger für solche Störungen sind, da sie nicht nur die Struktur der Daten, sondern auch das Rauschen lernen. Zur Bewertung der Robustheit werden laut Braiek und Khomh (2024) Eingabedaten gezielt verändert, um die Sensitivität des Modells gegenüber solcher Störungen zu testen. Bleiben die Ergebnisse dabei stabil, deutet dies laut der Autoren auf eine hohe Widerstandsfähigkeit gegenüber Unsicherheiten oder Abweichungen in den Daten hin. So ergänzt Robustheit die zuvor beschriebenen Metriken um eine wichtige praxisnahe Perspektive (Braiek & Khomh, 2024).

Ein gängiges Verfahren zur Schätzung der Leistungsfähigkeit von Klassifikationsmodellen ist die Kreuzvalidierung (Witten et al., 2017). Dabei wird der Datensatz in eine feste Anzahl von Teilmengen (Folds) zerlegt, die abwechselnd als Test- und Trainingsdaten verwendet werden (Witten et al., 2017). So wird in mehreren Durchläufen jeweils ein Fold zum Testen genutzt, während die übrigen Daten zum Training dienen, bis jedes Datensatzobjekt einmal als Testfall herangezogen wurde (Witten et al., 2017). Um Verzerrungen durch ungleich verteilte Klassen innerhalb der Folds zu vermeiden, werden die Folds in der Praxis häufig stratifiziert, also so gebildet, dass die Klassenverteilung in allen Folds möglichst den ursprünglichen Proportionen entspricht (Witten et al., 2017). Als Standardverfahren hat sich die stratifizierte zehnfache Kreuzvalidierung etabliert, da sie einen guten Kompromiss zwischen Rechenaufwand, Bias und Varianz bildet (Witten et al., 2017). Gleichzeitig zeigen aktuelle Studien, dass die konkrete Wahl der Fold-Anzahl einzelfallabhängig ist und kein fixes Optimum existiert, vielmehr kann bereits eine geringere Anzahl von Folds zu stabilen Ergebnissen führen und gleichzeitig den Rechenaufwand deutlich reduzieren (Mahlich et al., 2024). Die Ergebnisse der Kreuzvalidierung werden üblicherweise als Mittelwerte der gewählten Evaluationsmetriken ausgewertet und um deren Standardabweichung ergänzt, die Aufschluss über die Stabilität und Verlässlichkeit der Schätzung gibt (Kohavi, 1995). Damit stellt die Kreuzvalidierung ein etabliertes Verfahren dar, um die Aussagekraft von Evaluationsmetriken abzusichern und eine fundierte Einschätzung der Modellgüte zu ermöglichen.

In der Diskussion um Evaluationskriterien für Data-Mining-Verfahren gewinnt auch Fairness zunehmend an Bedeutung (Rebstadt et al., 2022). Fairness bezieht sich auf die Gleichbehandlung verschiedener Gruppen und die Vermeidung systematischer Verzerrungen (Rebstadt et al., 2022). Da im industriellen Kontext typischerweise keine personenbezogenen Merkmale verarbeitet werden, spielen ethische Kriterien wie Fairness bei der Auswahl und Bewertung von Data-Mining-Verfahren in dieser Arbeit keine Rolle.

Evaluationsmetriken ermöglichen die Beurteilung der Qualität und Leistungsfähigkeit von Data-Mining-Verfahren (Witten et al., 2017). Neben Accuracy, Precision, Recall und FM kommen je nach Problemstellung auch spezialisierte Metriken wie AUC oder hybride Verfahren zum Einsatz, insbesondere bei unausgeglichenen Datensätzen (Hossin & Sulaiman, 2015). Ergänzend zur Modellgüte ist auch die Effizienz relevant, etwa hinsichtlich Trainings- und Inferenzzeit (Schwartz et al., 2020). Weitere Bewertungskriterien wie die Robustheit eines Modells bei veränderten Eingaben (Braiek & Khomh, 2024) oder seine Komplexität (Bishop, 2006) liefern praxisnahe Hinweise zur Modellzuverlässigkeit. Die Kreuzvalidierung ist ein etabliertes Verfahren der Modellbewertung, da sie eine verlässliche Schätzung der Generalisierungsfähigkeit (Witten et al., 2017) sowie der Stabilität (Kohavi, 1995) ermöglicht. Damit Data-Mining-

Verfahren in der Praxis zuverlässig eingesetzt werden können, ist eine sorgfältige Vorbereitung der Daten unerlässlich (García et al., 2015). Der folgende Abschnitt stellt die grundlegenden Verfahren der Datenvorverarbeitung vor, die die Grundlage dafür bilden.

2.3 Einführung in die Datenvorverarbeitung

Wie im vorherigen Abschnitt dargelegt, bilden geeignete Evaluationsmetriken die Grundlage für die Beurteilung der Modellleistung. Damit ein Modell jedoch sinnvoll arbeiten kann, müssen die zugrunde liegenden Daten zuvor systematisch vorbereitet werden (García et al., 2015). Die Datenvorverarbeitung umfasst grundlegende und entscheidende Schritte, um Rohdaten für anschließende Analysen nutzbar zu machen (García et al., 2015). Daten liegen selten unmittelbar in einer Form vor, die direkte, valide und verlässliche Analysen erlaubt (García et al., 2015). In diesem Abschnitt werden daher jene Aspekte der Datenvorverarbeitung behandelt, die im industriellen Kontext besonders häufig auftreten und für die Qualität datengetriebener Analysen eine zentrale Rolle spielen. Dazu zählen unter anderem der Umgang mit Ausreißern, Rauschen und fehlenden Werten sowie Verfahren zur Datenintegration und -normalisierung, die in der Literatur als zentrale Aufgaben der Datenvorverarbeitung hervorgehoben werden (García et al., 2015).

Ausreißer sind Werte, die deutlich außerhalb des üblichen Bereichs eines Datensatzes liegen (Hodge & Austin, 2004). Sie entstehen bspw. durch Messfehler, Eingabefehler oder außergewöhnliche Ereignisse und können statistische Kennzahlen wie den Mittelwert stark beeinflussen (Hodge & Austin, 2004). Zur Erkennung eignen sich grafische Mittel wie Boxplots oder Streudiagramme, die ungewöhnlich hohe oder niedrige Werte visuell hervorheben (Hodge & Austin, 2004). Zusätzlich gibt es simple statistische Regeln zur Erkennung von Ausreißern. Eine dieser Regeln basiert auf dem 1,5-fachen Interquartilsabstand (IQR): Werte gelten als potenzielle Ausreißer, wenn sie mehr als das 1,5-Fache des IQR oberhalb des dritten oder unterhalb des ersten Quartils liegen (Hedderich & Sachs, 2016). Quartile unterteilen eine sortierte Datenreihe in vier gleich große Abschnitte: Das erste Quartil (Q1) ist der Wert, unter dem 25 % der Daten liegen, das dritte Quartil (Q3) der Wert, unter dem 75 % der Daten liegen (Hedderich & Sachs, 2016). Der IQR ist die Differenz zwischen Q3 und Q1 und stellt somit die Streuung der mittleren 50 % der Daten dar (Hedderich & Sachs, 2016). Für eine belastbare Interpretation der Analyseergebnisse ist es wichtig, erkannte Ausreißer angemessen zu behandeln (Hodge & Austin, 2004). Je nach Art der Analyse und Anforderungen an die Datengenauigkeit können sie entfernt, angepasst oder durch Verfahren berücksichtigt werden, die unempfindlich gegenüber extremen Werten sind (Hodge & Austin, 2004).

Neben Ausreißern kann auch Rauschen die Aussagekraft von Daten beeinträchtigen. Als Rauschen werden Schwankungen in Daten bezeichnet, die keinen systematischen Zusammenhang widerspiegeln (Han et al., 2012). Solche Störungen können bspw. bedingt durch Messungenauigkeiten oder Umwelteinflüsse auftreten und äußern sich als unregelmäßige Zacken oder Sprüngen in Datenvisualisierungen (Han et al., 2012). Diese zufälligen Abweichungen erschweren es, echte Muster oder Trends zu erkennen und können Analyseergebnisse verfälschen (Han et al., 2012). Zur Verringerung von Rauschen kommen verschiedene Verfahren zum Einsatz, darunter Glättungsverfahren wie der gleitende Durchschnitt (Han et al., 2012). Beim gleitenden Durchschnitt wird jeder Datenpunkt durch den Mittelwert seiner unmittelbaren Nachbarn ersetzt, um kurzfristige Schwankungen zu glätten (Han et al., 2012). Weitere Ansätze stammen aus der Signalverarbeitung, bspw. Filterverfahren, die hochfrequente Störungen unterdrücken (Han et al., 2012). Ziel dieser Maßnahmen ist es, zufällige Abweichungen zu reduzieren und die Struktur der Daten deutlicher sichtbar zu machen (Han et al., 2012).

Eine weitere Herausforderung bei der Datenvorverarbeitung ist die Integration von Daten aus mehreren Quellen (Rahm & Do, 2000). Daten stammen oft aus unterschiedlichen Quellen wie Sensoren, automatischen Protokollen (Log-Dateien) oder Datenbanken (Rahm & Do, 2000). Dies führt zu Redundanzen, Inkonsistenzen oder widersprüchlichen Werten (Rahm & Do, 2000). Ein häufiges Problem bei der Integration technischer Datenquellen besteht darin, dass

gleichartige Informationen in unterschiedlicher Struktur oder unter verschiedenen Bezeichnung vorliegen (Rahm & Do, 2000). So kann etwa dieselbe Produktkennung in einem System als Zeichenkette gespeichert sein, während sie in einem anderen als numerischer Code erscheint. Um diese strukturellen und semantischen Unterschiede zu beheben, kommen Verfahren wie Schema Matching zum Einsatz, das Ähnlichkeiten zwischen Datenfeldern identifiziert (Rahm & Do, 2000). Zusätzlich dient das sogenannte Record Linkage der Erkennung von Datensätzen, die sich auf dasselbe Objekt beziehen, jedoch aus verschiedenen Quellen stammen (Rahm & Do, 2000). Dabei werden bspw. zwei Einträge mit leicht abweichendem Namen als identisch erkannt und anschließend zu einem einheitlichen Datensatz konsolidiert (Rahm & Do, 2000). Ziel ist es, inhaltlich passende Einträge zu kombinieren und so einen konsistenten Datenbestand zu schaffen (Rahm & Do, 2000). Neuere Entwicklungen gehen dabei über diese Verfahren hinaus und setzen zunehmend auf maschinelle Lernansätze, etwa Schema-Matching-Netzwerke, die verschiedene Datenstrukturen simultan berücksichtigen und dadurch höhere Genauigkeiten ermöglichen (Rodrigues & Da Silva, 2021).

Sind die Daten strukturell vereinheitlicht, stellt sich im nächsten Schritt die Frage nach ihrer numerischen Vergleichbarkeit. Die Normalisierung oder Standardisierung von Daten wird notwendig, wenn Merkmale unterschiedliche Maßeinheiten oder Größenordnungen aufweisen (García et al., 2015). Ohne Anpassung könnten Merkmale mit großen Zahlenwerten die Analyse dominieren (García et al., 2015). Die Normalisierung bringt alle Werte in einen gemeinsamen Wertebereich, etwa zwischen 0 und 1 bei der Min-Max-Skalierung (García et al., 2015). Dabei wird der kleinste Wert eines Merkmals auf 0 und der Größte auf 1 gesetzt, alle anderen Werte werden entsprechend skaliert (García et al., 2015). Die Min-Max-Skalierung verhindert, dass Merkmale mit großen Wertebereichen dominieren (García et al., 2015). Dies ist insbesondere bei Verfahren problematisch, die auf Abständen oder Gewichtungen beruhen, wo die Größenordnung einzelner Merkmale die Ergebnisse stark beeinflussen kann (García et al., 2015). Bei der Standardisierung werden die Werte so angepasst, dass sie einen Mittelwert von 0 und eine Standardabweichung von 1 haben (Z-Transformation) (García et al., 2015). Diese Verfahren stellen sicher, dass alle Merkmale vergleichbar sind und gleich gewichtet werden (García et al., 2015).

In vielen realen Anwendungsfällen von Klassifikationsaufgaben treten außerdem stark unausgeglichene Klassenverteilungen auf, bei denen die Anzahl der Instanzen einer Klasse deutlich größer ist als die der anderen (Leevy et al., 2018). Ein solches Ungleichgewicht führt häufig dazu, dass ein Klassifikator überwiegend die Mehrheitsklasse vorhersagt, weil er aus dieser besonders viele Beispiele im Training gesehen hat (Leevy et al., 2018). Die Minderheitsklasse wird dadurch nur unzureichend erkannt, obwohl sie in vielen Anwendungen, wie bspw. beim Erkennen von Krankheiten, die eigentlich interessantere ist (Leevy et al., 2018). Um dieses Problem zu behandeln, werden Verfahren zum Ausgleich der Klassenverteilung verwendet. Dafür existieren verschiedene Resampling-Verfahren, die entweder Instanzen der Mehrheitsklasse entfernen (Random Under-Sampling) oder zusätzliche Instanzen der Minderheitsklasse erzeugen (Random Over-Sampling) (Leevy et al., 2018). Neben diesen einfachen Ansätzen gibt es auch komplexere Verfahren, wie die Synthetic Minority Over-sampling Technique (SMOTE), welche durch Interpolation zwischen bestehenden Instanzen generiert (Leevy et al., 2018). Konkret werden dazu für jede Instanz der Minderheitsklasse ihre k-nächsten Nachbarn herangezogen, wobei neue Datenpunkte entlang der Verbindungslinien zwischen diesen Beobachtungen gebildet werden (Chawla et al., 2002). Auf diese Weise erhält der Klassifikator mehr repräsentative Trainingsbeispiele der seltenen Klasse, wodurch die Abgrenzung der Klassen verbessert wird und die Erkennungsleistung für die Minderheitsklasse steigt (Chawla et al., 2002).

Neben diesen Aspekten gehören auch der Umgang mit fehlenden Werten und die Reduktion umfangreicher oder komplexer Datensätze zu den Kernaufgaben der Datenvorverarbeitung. Da diese Themen den Schwerpunkt der vorliegenden Arbeit bilden, werden sie in den folgenden Kapiteln vertiefend behandelt.

3 Umgang mit fehlenden Werten

Fehlende Werte sind ein häufiges und komplexes Problem in realen Datensätzen (Little & Rubin, 2019). Ihre Ursachen sind vielfältig und hängen meist mit defekten Sensoren, Netzwerkausfällen oder Systemwartungen in industriellen Systemen zusammen (Teh et al., 2020). Little und Rubin (2019, S.4) definieren fehlende Daten als „unobserved values that would be meaningful for analysis if observed". Diese Definition verdeutlicht, dass hinter einem fehlenden Wert möglicherweise eine für die Modellbildung und Interpretation relevante Information verborgen ist (Little & Rubin, 2019). Das Fehlen eines Wertes weist also auf eine Lücke im Datenverständnis hin, die, wenn sie nicht behoben wird, die Aussagekraft von Analysen laut Little und Rubin (2019) erheblich beeinträchtigen kann. Um mit fehlenden Werten angemessen umzugehen, ist es zunächst notwendig zu verstehen, auf welche Weise diese im Datensatz entstehen. Dabei spielt insbesondere die zugrundeliegende Mechanik des Fehlens eine Rolle, also die Frage, ob das Fehlen zufällig oder systematisch auftritt.

3.1 Typen fehlender Werte

Eine systematische Betrachtung der Ursachen fehlender Werte ermöglicht ihre Einteilung in verschiedene Typen. Nach Little und Rubin (2019) lassen sich fehlende Werte in folgenden drei Hauptkategorien einteilen.

1. Missing Completely at Random (MCAR): Das Fehlen von Daten erfolgt völlig zufällig und unabhängig von allen beobachteten und unbeobachteten Variablen (Little & Rubin, 2019). Daher besteht keine systematische Beziehung zwischen fehlenden Werten und anderen Werten im Datensatz (Little & Rubin, 2019). MCAR liegt bspw. vor, wenn Messwerte aufgrund eines temporären Sensorausfalls fehlen, der nicht mit anderen Variablen zusammenhängt.
2. Missing at Random (MAR): Das Fehlen der Daten hängt mit anderen beobachteten Variablen zusammen, nicht jedoch mit den fehlenden Werten selbst (Little & Rubin, 2019). Das Fehlen eines Wertes kann durch andere, im Datensatz vorhandene Merkmale erklärt werden (Little & Rubin, 2019). Nach Fleck et al. (2020) tritt dieser Fall in industriellen Kontexten häufig auf, wenn bestimmte Sensoren entlang unterschiedlicher Produktionsrouten nicht aktiv sind und somit keine Daten liefern, während andere Sensoren weiterhin Messwerte erfassen.
3. Missing Not at Random (MNAR): Hier hängt das Fehlen von Daten direkt mit den fehlenden Werten selbst zusammen (Little & Rubin, 2019). Ein Beispiel für diesen Fall ist das bewusste Nichtbeantworten sensibler Fragen in Umfragen oder der Abbruch einer klinischen Studie durch besonders stark betroffene Patienten (Little & Rubin, 2019).

Diese Unterscheidung ist essenziell, da sie die Auswahl geeigneter Verfahren zur Behebung maßgeblich bestimmt und entscheidenden Einfluss auf die Validität der Analyseergebnisse hat (Little & Rubin, 2019). Besonders im industriellen Umfeld, in dem Ausfälle oder Lücken oft systematisch auftreten, ist die genaue Analyse des Ausfallmusters eine notwendige Voraussetzung für eine methodisch fundierte Datenvorverarbeitung (Fleck et al., 2020). Im nächsten Abschnitt wird daher zunächst die Erkennung fehlender Werte thematisiert, bevor geeignete Verfahren zur Behandlung vorgestellt werden.

3.2 Verfahren zur Erkennung fehlender Werte

Im Rahmen der explorativen Datenanalyse (EDA) können auch fehlende Werte sichtbar gemacht werden. Die EDA ist ein methodischer Ansatz zur Untersuchung und Visualisierung von Datensätzen, bevor komplexere Analysen durchgeführt werden (Cleff, 2015). Zu den zentralen Elementen der EDA gehören grafische Darstellungen wie Histogramme, Boxplots und Streudiagramme, die die allgemeine Identifikation von Anomalien und Mustern erleichtern (Cleff,

2015). Ergänzt wird dies durch deskriptive Statistiken wie Mittelwert, Median und Quartile, die erste Einblicke in die Verteilung und Struktur der Daten ermöglichen (Cleff, 2015). Zur Identifikation und Analyse fehlender Werte existieren verschiedene Verfahren der EDA, die im Folgenden vorgestellt werden.

Heatmaps

Matrix Plots (Heatmaps) sind visuelle Darstellungen, die fehlende Daten durch farbliche Markierungen kennzeichnen und so deren Verteilung im Datensatz deutlich machen (Fernstad, 2019). Durch die Verwendung von Heatmaps können insbesondere systematische Muster oder Cluster fehlender Werte identifiziert werden (Fernstad, 2019). Dieses Verfahren ermöglicht es, auch größere Bereiche mit fehlenden Werten im Datensatz auf einen Blick zu erkennen (Fernstad, 2019). Auch wiederkehrende Muster fehlender Werte über mehrere Variablen hinweg werden durch die Visualisierung deutlich sichtbar (Fernstad, 2019).

Missingness Matrix

Die Missingness Matrix ist eine spezielle Matrixdarstellung, die für jede Beobachtung und jede Variable anzeigt, ob ein Wert vorhanden oder fehlend ist (Little & Rubin, 2019). Durch die Strukturierung dieser Matrix können Ausfallmuster sichtbar gemacht werden (Little & Rubin, 2019). Auf diese Weise lassen sich systematisch fehlende Datenbereiche identifizieren, die auf unterschiedliche Ursachen hindeuten können (Little & Rubin, 2019). Die Missingness Matrix bildet damit eine zentrale Grundlage für die Analyse von Ausfallmustern (Little & Rubin, 2019). In der Literatur wird sie auch als Shadow Matrix bezeichnet und dient dort als flexible Grundlage, um das Verhalten fehlender Werte zu explorieren und Abhängigkeiten sichtbar zu machen (Tierney & Cook, 2023). Die folgende Abbildung 3.1 zeigt das Prinzip der Missingness Matrix.

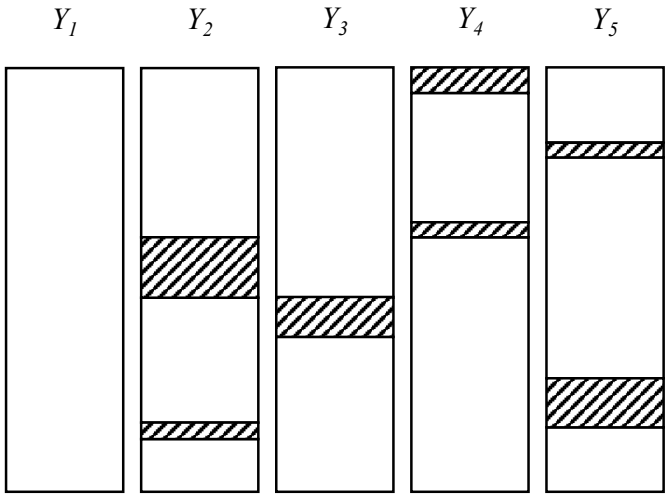


Abbildung 3.1: Missingness Matrix in Anlehnung an Little und Rubin (2019, S. 9)

Die in Abbildung 3.1 abgebildete Missingness Matrix zeigt für jede Beobachtung und jede Variable (Y) an, ob ein Wert vorhanden oder fehlend ist. Nicht schraffierte Flächen stehen dabei für vorhandene Daten, während schraffierte Bereiche fehlende Daten lokalisieren. Auf diese Weise lassen sich systematische Muster wie blockweise fehlende Daten oder zufällige Ausfälle erkennen. Die visuelle Darstellung ermöglicht einen schnellen Überblick über die Struktur des Datensatzes und erleichtert die Identifikation auffälliger Datenlücken.

Statistische Tests

Neben visuellen Darstellungen besteht ebenfalls die Möglichkeit, fehlende Daten mithilfe von Statistischen Tests zu erkennen (Little & Rubin, 2019). Mit dem MCAR-Test, der auf dem Vergleich von Verteilungen der vorhandenen und fehlenden Daten basiert, lässt sich prüfen, ob

fehlende Daten vollständig zufällig verteilt sind (Little, 1988). Bleiben die Unterschiede statistisch insignifikant, ist davon auszugehen, dass die Daten MCAR, also unabhängig von anderen beobachteten oder unbeobachteten Variablen fehlen (Little, 1988). Weichen die Verteilungen jedoch signifikant voneinander ab, deutet dies darauf hin, dass die Annahme von MCAR nicht zutrifft (Little, 1988). In solchen Fällen ist das Ausfallmuster möglicherweise durch andere Variablen erklärbar, was auf MAR oder MNAR hindeuten würde (Little & Rubin, 2019). Der MCAR-Test nach Little (1988) stößt allerdings insbesondere in hochdimensionalen Datensätzen an seine Grenzen, da seine Verteilungseigenschaften dort oft nicht zuverlässig gelten (Bordino & Berrett, 2024). Bordino und Berrett (2024) schlagen deshalb bootstrap-basierte Verfahren vor, die die Verteilung der Teststatistik durch Resampling approximieren und so auch ohne starke Verteilungsannahmen robuste Ergebnisse ermöglichen. Auf diese Weise lässt sich die Gültigkeit der MCAR-Annahme auch in komplexen, hochdimensionalen Datenssätzen verlässlicher prüfen.

Die Identifikation fehlender Werte und die Analyse ihrer Verteilung sind grundlegende Schritte, um geeignete Behandlungsstrategien auszuwählen. Im folgenden Abschnitt werden verschiedene Verfahren zur Behandlung fehlender Werte vorgestellt, die sich je nach Art des Fehlens und Datenstruktur unterscheiden.

3.3 Verfahren zur Behandlung fehlender Werte

Die Wahl eines angemessen Verfahrens zur Behandlung fehlender Werte hängt, wie in Abschnitt 3.1 beschrieben, vom Mechanismus des Fehlens der Daten ab. Bei der Auswahl geeigneter Verfahren spielen insbesondere Überlegungen zur Verzerrung und Effizienz der Schätzer eine zentrale Rolle (Little & Rubin, 2019). Eine grundlegende Unterscheidung im Umgang mit fehlenden Werten besteht laut Little und Rubin (2019) zwischen Verfahren, die unvollständige Beobachtungen ausschließen, sogenannte Eliminierungsverfahren, und Verfahren, die fehlende Werte durch plausible Ersatzwerte ersetzen, den sogenannten Imputationsverfahren. Diese Einteilung bildet die Basis vieler methodischer Ansätze zur Behandlung von fehlenden Werten (Little & Rubin, 2019). In diesem Kapitel werden die einzelnen Verfahrenskategorien separat vorgestellt. Dabei liegt der Fokus neben der Beschreibung der jeweiligen Verfahren auch auf ihrer Bewertung im Hinblick auf Voraussetzungen, Eigenschaften und mögliche Konsequenzen der Anwendung. Im ersten Schritt werden die Eliminierungsverfahren betrachtet, die zur Entfernung unvollständiger Beobachtungen verwendet werden.

3.3.1 Eliminierungsverfahren

Eliminierungsverfahren umfassen Verfahren zur Behandlung unvollständiger Daten- oder Distanzmatrizen, bei denen Objekte oder Merkmale mit fehlenden Werten aus der Analyse ausgeschlossen werden (Bankhofer, 1995). Diese Verfahren sind allerdings nur uneingeschränkt anwendbar, wenn das Fehlen der Daten MCAR und somit unabhängig von den untersuchten Variablen ist (Little & Rubin, 2019). Da keine weiteren Anforderungen an die Verteilung der Daten oder das Skalenniveau gestellt werden, können diese Verfahren grundsätzlich bei beliebigen Datentypen eingesetzt werden (Bankhofer, 1995). Bei unmittelbar erhobenen Distanzmatrizen beschränkt sich die Anwendung jedoch auf die Eliminierung von Objekten (Bankhofer, 1995). Zunächst werden die Verfahren der Objekteliminierung näher betrachtet.

Objekteliminierung

Wenn die Merkmalsstruktur untersucht werden soll, kann das Ausschließen von Objekten mit fehlenden Werten eine geeignete Lösung darstellen (Beaulieu-Jones et al., 2018). Die sogenannte Objekteliminierung ist bei der Analyse von Ähnlichkeitsstrukturen jedoch nur sinnvoll, wenn nur wenige Objekte betroffen sind (Bankhofer, 1995).

Complete-Case-Analysis bezeichnet ein Verfahren der Objekteliminierung (Bankhofer, 1995). Im Rahmen der Complete-Case-Analysis werden sämtliche Objekte eines Datensatzes entfernt bzw. aus der Analyse ausgeschlossen, deren Merkmalsausprägungen unvollständig

sind (Bankhofer, 1995). Dieses Verfahren ist einfach in der Anwendung, kann jedoch zu einem erheblichen Informationsverlust führen, insbesondere bei großen Datenlücken (Bankhofer, 1995).

Available-Case-Analysis ist im Gegensatz zur Complete-Case-Analysis ein Verfahren, bei dem für jede Analyse nur der verfügbare Teil der Daten verwendet wird (Bankhofer, 1995). Das bedeutet, dass bei der Berechnung einer univariaten Statistik wie bspw. dem Mittelwert alle vorhandenen Werte eines Merkmals einbezogen werden, wodurch die Stichprobengröße je nach Variable unterschiedlich ist (Bankhofer, 1995). Dieser Umstand kann nach Bankhofer (1995) jedoch die Vergleichbarkeit der Ergebnisse beeinträchtigen.

Abbildung 3.2 zeigt basierend auf Bankhofer (1995) die Auswirkungen beider Analyseverfahren auf einen Datensatz mit fehlenden Werten. Die Darstellung verdeutlicht, wie sich die Verfahren hinsichtlich der verbleibenden Daten unterscheiden.

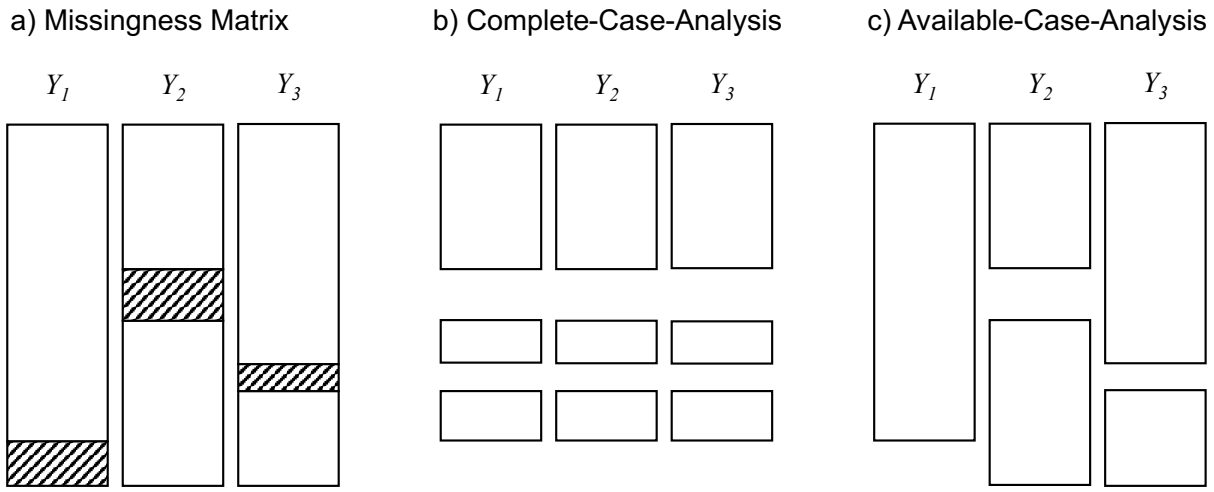


Abbildung 3.2: Vergleich von Complete-Case-Analysis (b) und Available-Case-Analysis (c) anhand einer Missingness Matrix (a)

Abbildung 3.2 setzt sich aus drei verschiedenen Grafiken zusammen. In (a) ist die ursprüngliche Missingness Matrix dargestellt, wobei schraffierte Felder fehlende Werte markieren. In (b) werden bei der Complete-Case-Analysis alle Beobachtungen ausgeschlossen, die mindestens einen fehlenden Wert aufweisen. In (c) zeigt die Available-Case-Analysis, dass alle vorhandenen Werte in die Analyse einbezogen werden, auch wenn einzelne Werte innerhalb eines Beobachtungsfalls fehlen. Neben der Objekteliminierung gibt es auch die Merkmalseliminierung, die nachfolgend näher betrachtet wird.

Merkmalseliminierung

Die Merkmalseliminierung bezeichnet den Ausschluss von Variablen mit fehlenden Werten aus der Analyse (Bankhofer, 1995). Auch wenn dieser Ansatz inhaltlich von der Objekteliminierung zu unterscheiden ist, folgt er einem vergleichbaren formalen Prinzip (Bankhofer, 1995). Der Autor unterscheidet hier nach Complete- und Available-Variable-Analysis.

Complete-Variable-Analysis bezeichnet ein Verfahren, bei dem ausschließlich die Merkmale in die Auswertung einbezogen werden, für die vollständige Werte über alle Objekte vorliegen (Bankhofer, 1995). Die unvollständigen Merkmale werden aus der Analyse ausgeschlossen, wodurch sich die Datenmatrix auf eine kleinere Dimension reduziert (Bankhofer, 1995). Ziel ist es die nachfolgenden Analysen anhand einer vollständig gefüllten Teilmatrix durchzuführen, die jedoch weniger Merkmale enthält als die ursprüngliche Datenmatrix (Bankhofer, 1995). Ein Vorteil dieses Verfahrens besteht darin, dass alle Objekte der verbleibenden Merkmale erhalten bleiben, was für Analysen zur Ähnlichkeitsstruktur oft günstiger ist (Wishart, 1978). Gleich-

zeitig besteht jedoch die Gefahr, dass durch den Ausschluss einzelner Merkmale wichtige Informationen verloren gehen, da diese Merkmale nicht zufällig, sondern meist gezielt erhoben wurden (Evans et al., 1979).

Available-Variable-Analysis hingegen bezeichnet das Durchführen von Analysen ausschließlich auf den jeweils verfügbaren Merkmalswerten (Bankhofer, 1995). Dabei wird für jede Berechnung nur auf die Merkmale zurückgegriffen, die für die jeweils betrachteten Objekte tatsächlich vorliegen (Bankhofer, 1995). Die Analyse erfolgt also individuell auf Grundlage der jeweils verfügbaren Werte, ohne dass ganze Merkmale aus dem Datensatz ausgeschlossen werden müssen (Bankhofer, 1995). Dadurch kann ein größerer Teil der Daten genutzt werden, allerdings variiert die Berechnungsgrundlage je nach Objektpaar (Bankhofer, 1995). Da jeweils nur die gemeinsam vorhandenen Merkmale eines Objektpaares berücksichtigt werden, kann trotz fehlender Werte für jede Kombination ein Distanzwert berechnet werden (Bankhofer, 1995). Auf diese Weise entsteht eine vollständige Distanzmatrix (Bankhofer, 1995). Die folgende Tabelle 3.1 stellt die die Voraussetzungen und Eigenschaften der vier betrachteten Eliminierungsverfahren gegenüber.

Tabelle 3.1: Vergleich der Eliminierungsverfahren in Anlehnung an Bankhofer (1995, S. 102)

Verfahren	Voraussetzungen	Eigenschaften
Complete-Case-Analysis	Daten sind MCAR	ggf. erheblicher Informationsverlust; Analyse der Ähnlichkeitsstruktur aller Objekte nicht mehr möglich
Available-Case-Analysis	Daten sind MCAR	kein bzw. geringer Informationsverlust; variierender Stichprobenumfang
Complete-Variable-Analysis	Daten sind MCAR	ggf. erheblicher Informationsverlust; alle Objekte der verbleibenden Merkmale bleiben erhalten (günstig für Analysen zur Ähnlichkeitsstruktur)
Available-Variable-Analysis	Daten sind MCAR	kein bzw. geringer Informationsverlust; Ergebnis ist eine vollständige, mit geeigneten Verfahren auswertbare Distanzmatrix

Tabelle 3.1 zeigt, dass alle vier betrachteten Verfahren voraussetzen, dass die fehlenden Daten MCAR sind, also vollständig zufällig auftreten. Während die Complete-Case-Analysis und die Complete-Variable-Analysis zu erheblichem Informationsverlust führen können, erlauben die Available-Case-Analysis und Available-Variable-Analysis eine flexiblere Nutzung der vollständigen Daten (Bankhofer, 1995). Letztere Varianten beziehen jeweils die tatsächlich vorhandenen Werte in Analysen ein und können auch so bei unvollständigen Datensätzen aussagekräftige Ergebnisse liefern (Bankhofer, 1995).

In vielen Fällen ist das Entfernen fehlender Werte nicht sinnvoll oder praktikabel, da es zu einem erheblichen Informationsverlust führen kann, insbesondere wenn die Annahme von MCAR nicht zutrifft (Little & Rubin, 2019). Ein alternativer Umgang besteht darin, die fehlenden Werte auf Grundlage der vorhandenen Daten zu ergänzen, also zu imputieren (Bankhofer, 1995). Daher erfolgt im nächsten Abschnitt eine Einführung in grundlegenden Imputationsverfahren.

3.3.2 Einführung in die Imputationsverfahren

Nach Bankhofer (1995) sind Imputationsverfahren solche Verfahren, bei denen fehlende Werte durch Schätzungen ersetzt werden, die auf den vorhandenen Daten basieren. Durch den Einsatz geeigneter Imputationsverfahren können die ursprüngliche Struktur und Größe

des Datensatzes erhalten werden, wodurch insbesondere Verzerrungen der Varianzen und Kovarianzen vermieden werden können (Little & Rubin, 2019). Die Verwendung von Imputationsverfahren setzt dabei voraus, dass die fehlenden Daten einem zugrunde liegenden Mechanismus folgen, der als MAR oder MCAR klassifiziert wird, da bei diesen Mechanismen die Wahrscheinlichkeit des Fehlens nicht von den fehlenden Werten selbst abhängt und somit eine verlässliche Schätzung möglich ist (Schafer & Graham, 2002). Ist diese Voraussetzung nicht erfüllt und das Fehlen der Daten hängt direkt mit den fehlenden Werten selbst zusammen (MNAR), können fehlerhafte oder verzerrte Ergebnisse entstehen (Little & Rubin, 2019). Eine Berücksichtigung dieser Voraussetzung trägt dazu bei, Verzerrungen in den Analyseergebnissen zu minimieren (Bankhofer, 1995). Allerdings sollte bedacht werden, dass es sich bei imputierten Werten, auch wenn sie unverzerrte Ergebnisse liefern, stets um Schätzungen der realen, unbekannten Werte handelt, die nicht mit den tatsächlich vorhanden Werten verwechselt werden dürfen (Madow et al., 1983).

Grundlegend lässt sich die Imputation fehlender Werte in einfache oder statistische Imputationsverfahren, Klassenimputation sowie multivariate und modellbasierte Imputationsverfahren unterteilen (Bankhofer, 1995). Statistische Imputationsverfahren wie die Mittelwert- oder Medianimputation ersetzen fehlende Werte durch grundlegende statistische Kenngrößen (Bankhofer, 1995). Komplexere Verfahren hingegen verwenden aufgrund ihrer Nutzung multivariater Zusammenhänge Informationen aus der gesamten Datenmatrix (Bankhofer, 1995) oder nutzen statistische und maschinelle Lernverfahren (Jerez et al., 2010). Gerade die Verfahren der multiplen Imputation gelten in der Fachliteratur als weit verbreitet und etabliert, da sie durch die wiederholte Schätzung der fehlenden Werte die Unsicherheit der Imputation explizit berücksichtigen und somit robustere Schätzungen der Modellparameter liefern (Schafer & Graham, 2002). In jüngerer Zeit haben auch Verfahren auf Basis maschineller Lernalgorithmen wie neuronale Netzwerke, Entscheidungsbäume und kNN-Imputationen aufgrund ihrer Fähigkeit, komplexe, nichtlineare Zusammenhänge in Datensätzen zu modellieren, zunehmend an Bedeutung gewonnen (Jerez et al., 2010).

Damit steht eine Vielzahl von Ansätzen zur Verfügung, die sich sowohl hinsichtlich ihrer theoretischen Fundierung als auch ihrer praktischen Einsatzmöglichkeiten unterscheiden. Für ein fundiertes Verständnis ist es daher erforderlich, die wichtigsten Verfahren systematisch vorzustellen, ihre jeweiligen Stärken und Schwächen herauszuarbeiten sowie ihre Eignung für unterschiedliche Datenkontexte zu bewerten. Zunächst werden die statistischen Imputationsverfahren betrachtet, welche durch ihre einfache Anwendbarkeit gekennzeichnet sind.

3.3.3 Statistische Imputationsverfahren

Statistische Imputationsverfahren beinhalten nach Bankhofer (1995) die Ersetzung fehlender Werte durch grundlegende statistische Kenngrößen oder feste Regeln. Der Autor beschreibt diese Verfahren als besonders unkompliziert in der Anwendung, da sie keine komplexen Modellannahmen oder iterative Berechnungen erfordern. In der Praxis finden statistische Imputationsverfahren insbesondere bei explorativen Studien oder bei geringem Umfang fehlender Daten Anwendung, da sie einfach umzusetzen und recheneffizient sind (García et al., 2015).

Statistisch problematisch ist jedoch, dass das wiederholte Einsetzen statistischer Kenngrößen dazu führen kann, dass die ursprüngliche Streuung in den Daten reduziert wird (Schafer & Graham, 2002). Dadurch wird die Varianz unterschätzt und die Datenverteilung künstlich zentralisiert (Schafer & Graham, 2002). Zusätzlich wird die Kovarianzstruktur zwischen Variablen verzerrt, da alle imputierten Werte identisch sind und somit keine natürliche Variation enthalten (Little & Rubin, 2019). Solche Verzerrungen können insbesondere dann zu erheblichen Fehleinschätzungen führen, wenn statistische Analyseverfahren stark auf die Erhaltung von Varianz- und Kovarianzstrukturen angewiesen sind (Little & Rubin, 2019). Im Folgenden werden die wichtigsten statistischen Imputationsverfahren hinsichtlich ihrer Funktion beschrieben.

Mittelwertimputation

Die Mittelwertimputation ist ein häufig angewandtes Verfahren, bei dem fehlende Werte durch den arithmetischen Mittelwert der beobachteten Werte ersetzt werden (Little & Rubin, 2019). Der Vorteil dieses Verfahrens liegt in seiner einfachen Implementierung (García et al., 2015). Andererseits kommen bereits beschriebene Verzerrungseffekte aufgrund wiederholten Einsetzens identischer Werte zum Tragen (Little & Rubin, 2019).

Medianimputation

Die Medianimputation basiert auf demselben Prinzip, nutzt jedoch den Median als Imputationswert (García et al., 2015). Gegenüber der Mittelwertimputation bietet sie einen Vorteil in Datensätzen mit schiefen Verteilungen oder Ausreißern, da der Median als robuste Lagekennzahl empfindlich gegenüber Extremwerten ist (García et al., 2015). Auch bei der Medianimputation werden jedoch Varianz und Kovarianz systematisch verfälscht, insbesondere wenn viele fehlende Werte durch den Median ersetzt werden (García et al., 2015).

Modusimputation

Die Modusimputation wird insbesondere bei nominalskalierten Variablen angewendet, also bei Merkmalen, deren Ausprägungen lediglich unterschiedliche Kategorien ohne natürliche Rangfolge darstellen (García et al., 2015). Hierbei wird der häufigste beobachtete Wert, der Modus, als Ersatz für fehlende Werte eingesetzt (García et al., 2015). Dieses Verfahren ist einfach anzuwenden und erfordert ebenfalls keine komplexen Annahmen oder Zusatzinformationen, kann jedoch zu einer Überrepräsentation der dominanten Kategorie führen (García et al., 2015). Dies ist vor allem in Klassifikationsaufgaben problematisch, in denen die relative Verteilung der Kategorien eine Rolle spielt (Bankhofer, 1995). In solchen Fällen kann der dominanten Kategorie nach Anwendung des Verfahrens ein überproportional hoher Anteil zugewiesen werden, was die Aussagekraft kategorialer Merkmale verzerrt (Bankhofer, 1995).

Konstantenimputation

Die Konstantenimputation ersetzt fehlende Werte durch einen vordefinierten, häufig arbiträren Wert (García et al., 2015). Als arbiträr gelten Werte, die ohne Bezug zu den statistischen Eigenschaften der zugrunde liegenden Datenstruktur gewählt werden (García et al., 2015), wie etwa 0 oder ein Textstring wie „unbekannt“. Dieses Verfahren ist insbesondere in der praktischen Datenaufbereitung verbreitet, etwa als Vorbereitungsschritt für maschinelle Lernverfahren, die keine fehlenden Werte verarbeiten können (García et al., 2015). Aus statistischer Sicht ist diese Vorgehensweise jedoch problematisch, da die eingefügten Werte keine validen Beobachtungen darstellen und in Analysen als eigenständige Ausprägung interpretiert werden (García et al., 2015). In Entscheidungsbäumen oder Clustering-Algorithmen kann dies zu falsch gewichteten Teilgruppen führen, da der konstant eingesetzte Wert vom Algorithmus als regulärer Datenpunkt interpretiert werden kann (García et al., 2015). Dadurch entstehen künstliche Klassen oder Gruppen, die lediglich auf der Präsenz eines Imputationswerts beruhen (García et al., 2015).

LOCF/NOCB

Besondere Varianten einfacher Imputationsverfahren sind die Verfahren Last Observation Carried Forward (LOCF) und Next Observation Carried Backward (NOCB), die typischerweise bei zeitbezogenen Daten zum Einsatz kommen (Jahangiri et al., 2023). LOCF ersetzt einen fehlenden Wert durch den letzten beobachteten Wert desselben Merkmals für dieselbe Untersuchungseinheit (Jahangiri et al., 2023). NOCB funktioniert analog, indem es den nächsten bekannten Wert in rückwärtiger Richtung verwendet (Jahangiri et al., 2023). Die Funktionsweise beider Verfahren ist in der folgenden Abbildung 3.3 anschaulich dargestellt.

Abbildung 3.3 zeigt exemplarisch die Anwendung der Verfahren LOCF und NOCB zur Imputation fehlender Temperaturdaten in einem Zeitreihendatensatz. In (a) ist der ursprüngliche Datensatz dargestellt, in dem für zwei aufeinanderfolgende Zeitpunkte keine Temperaturdaten

a) Ausgangsdatensatz

Zeit t	Temperatur T (°C)
08:00:00	72,4
08:01:00	-
08:02:00	-
08:03:00	73,1

b) LOCF

Zeit t	Temperatur T (°C)
08:00:00	72,4
08:01:00	72,4
08:02:00	72,4
08:03:00	73,1

c) NOCB

Zeit t	Temperatur T (°C)
08:00:00	72,4
08:01:00	73,1
08:02:00	73,1
08:03:00	73,1

Abbildung 3.3: Anwendungsbeispiel für die Imputationsverfahren LOCF (b) und NOCB (c) am Ausgangsdatensatz (a)

vorliegen. In (b) wird die Imputation mithilfe von LOCF verdeutlicht, bei der der zuletzt beobachtetet Wert (72,4°C) verwendet wird, um die darauffolgenden fehlenden Werte aufzufüllen. In (c) ist das Verfahren NOCB dargestellt, bei dem stattdessen der nächste verfügbare Wert (73,1°C) rückwirkend in die vorhergehenden Leerstellen imputiert wird.

Beide Verfahren setzen voraus, dass sich der Wert über die Zeit nicht wesentlich verändert, was jedoch in vielen realen Anwendungssituationen nicht haltbar ist (Jahangiri et al., 2023). Wird unter dieser Annahme imputiert, obwohl sich die tatsächlichen Werte über die Zeit signifikant verändern, kann dies zu verzerrten Analyseergebnissen führen (Jahangiri et al., 2023).

Zusammengefasst sind statistische Imputationsverfahren also insbesondere dann gerechtfertigt, wenn der Anteil fehlender Werte gering ist und keine komplexen Analysen durchgeführt werden sollen (García et al., 2015). In solchen Fällen bieten sie eine pragmatische Möglichkeit zur Datenvervollständigung bei gleichzeitig geringem Aufwand (Bankhofer, 1995). Little und Rubin (2019) beschreiben statistische Imputationsverfahren als zwar verführerisch aufgrund ihrer einfachen Anwendung, zugleich aber auch als potenziell stark irreführend. Für anspruchsvollere Analysen, bei denen eine realitätsnahe Rekonstruktion der Datenstruktur erforderlich ist, sollten jedoch Verfahren bevorzugt werden, die auch die Unsicherheit der Imputationen statistisch berücksichtigen (Schafer & Graham, 2002).

3.3.4 Klassenbasierte Imputationsverfahren

Die Klassenimputation basiert auf dem Ansatz, fehlende Werte nicht anhand aller verfügbarer Beobachtungen zu ersetzen, sondern gezielt innerhalb kleiner Gruppen (Bankhofer, 1995). Diese Teilgruppen, sogenannte Imputationsklassen, beinhalten Beobachtungen mit ähnlichen Eigenschaften und sollen eine realistischere Imputation erzielen (Bankhofer, 1995). Die Klassen werden anhand verfügbarer, vollständig beobachtbarer Merkmale definiert, um strukturell vergleichbare Beobachtungen zusammenzufassen (Bankhofer, 1995). Innerhalb jeder Klasse wird für jede fehlende Merkmalsausprägung eine passende Beobachtung mit vollständig vorhandenem Wert ausgewählt (Bankhofer, 1995). Der entsprechende Wert dieses sogenannten Spenderfalls dient anschließend als Imputationswert (Bankhofer, 1995). Ziel dieser Vorgehensweise ist es, die Heterogenität innerhalb der Daten zu berücksichtigen und dadurch plausiblere Ersetzungen zu erzielen, da die Ersetzung innerhalb homogener Gruppen auf strukturell ähnlichen Fällen basiert (Bankhofer, 1995). Im Folgenden werden für ein besseres Verständnis einige klassenbasierte Imputationsverfahren näher betrachtet.

Cold-Deck-Imputation

Ein grundlegendes Verfahren der Klassenimputation ist die Cold-Deck-Imputation (Bankhofer, 1995). Dabei erfolgt die Ersetzung fehlender Werte auf Grundlage von Daten aus einer externen Quelle (Little & Rubin, 2019), welche in keinem direkten Zusammenhang mit den Daten aus dem zu behandelnden Datensatz steht (Bankhofer, 1995). Diese Quelle kann bspw. eine frühere Erhebung von Daten im gleichen Kontext oder eine Zusatzstudie sein (Bankhofer,

1995). Die Imputationswerte werden innerhalb zuvor gebildeter Imputationsklassen gebildet (Bankhofer, 1995). Dabei wird vorausgesetzt, dass die externen Daten den gleichen Klassifikationsmerkmalen unterliegen wie der aktuelle Datensatz (Chapman, 1976). Im Cold-Deck-Verfahren wird jedem Fall innerhalb einer Imputationsklasse derselbe, zuvor definierte Ersatzwert zugewiesen, die Zuordnung erfolgt demnach deterministisch (Schnell, 1986). Aufgrund der Abhängigkeit von externen Datenquellen und der damit verbundenen methodischen Anforderungen gelten Cold-Deck-Verfahren insgesamt als wenig praxistauglich und werden in datenanalytischen Anwendungen nur selten genutzt (Bankhofer, 1995).

Hot-Deck-Imputation

Die Hot-Deck-Imputation stellt eine weitere Form der Klassenimputation dar (Schnell, 1986). Eine einheitliche Definition existiert nicht, doch zentrale Merkmale sind die Verwendung von Daten aus dem aktuellen Datensatz zur Ersetzung fehlender Werte sowie der sogenannte Dopplungscharakter, bei dem Werte existierender Beobachtungen auf andere übertragen werden (Schnell, 1986). Weil es naheliegender ist, fehlende Werte durch die Merkmalsausprägungen eines ähnlichen Falles zu ersetzen statt durch solche eines beliebig gewählten, erfolgt die Imputation im Rahmen der Hot-Deck-Verfahren innerhalb zuvor definierter Imputationsklassen (Schnell, 1986). Diese Vorgehensweise basiert auf den charakteristischen Merkmalen des Verfahrens, wie sie unter anderem von Ford (1983) beschrieben wurden:

1. Festlegung der Imputationsklassen
2. Auswahl einer vorhandene Ausprägung bezüglich desselben Merkmals innerhalb der Klassen für jede fehlende Merkmalsausprägung
3. Imputation der fehlenden Daten durch die jeweils ausgewählten Werte

Bei der Imputation innerhalb zuvor gebildeter Klassen unterscheidet Schnell (1986) zwischen sequenziellen und simultanen Verfahren. Sequenzielle Verfahren durchlaufen den Datensatz schrittweise und ersetzen fehlende Werte durch den jeweils zuletzt in der Klasse beobachteten gültigen Wert (Schnell, 1986). Simultane Verfahren hingegen suchen für jeden fehlenden Wert gezielt einen Spenderfall, der den geringsten Abstand in den relevanten Klassifikationsmerkmalen aufweist, wobei dieser Abstand über ein Distanzmaß quantifiziert wird (Schnell, 1986). Klassenbasierte Imputationsverfahren bieten eine differenzierte Ersetzung fehlender Werte innerhalb homogener Gruppen (Bankhofer, 1995). Bei zunehmender Komplexität der Daten stoßen sie jedoch an methodische Grenzen, bspw. wenn multivariate Abhängigkeiten zwischen Variablen nicht ausreichend berücksichtigt werden können (Schafer & Graham, 2002). Im Anschluss werden multivariate Imputationsverfahren betrachtet, die in solchen Fällen besser geeignet sind.

3.3.5 Multivariate Imputationsverfahren

Multivariate Imputationsverfahren zeichnen sich dadurch aus, dass sie bei der Ersetzung fehlender Werte mehrere Variablen gleichzeitig berücksichtigen (Schafer & Graham, 2002). Anders als bisher betrachtete einfache oder klassenbasierte Verfahren, die sich auf einzelne Merkmale stützen, modellieren multivariate Ansätze die gemeinsame Verteilung der beobachteten Daten und nutzen diese, um plausible Werte zu generieren (Little & Rubin, 2019). Ziel multivariater Imputationsverfahren ist es, Struktur, Zusammenhänge und Varianz des Datensatzes möglichst vollständig zu erhalten (Little & Rubin, 2019). Insbesondere bei komplexen Datensätzen mit vielen korrelierten Variablen bieten multivariate Verfahren eine statistisch fundierte Grundlage für die Imputation (Jerez et al., 2010). Ein häufig angewandtes Verfahren ist die Multiple Imputation, bei der für jede fehlende Beobachtung mehrere plausible Werte generiert werden, um Unsicherheit explizit zu berücksichtigen (Schafer & Graham, 2002). Verfahren der multivariaten Imputation setzen in der Regel voraus, dass die Daten MCAR oder MAR sind, da andernfalls die Schätzung verzerrt sein könnte (Little & Rubin, 2019). Im Folgenden wird ein Überblick über die wichtigsten multivariaten Verfahren gegeben, die sich hinsichtlich ihrer theoretischen Grundlagen und Einsatzbereiche unterscheiden.

Multiple Imputation

Die Multiple Imputation ist ein Verfahren, welches die Unsicherheit bei der Ersetzung fehlender Werte explizit berücksichtigt (Schafer & Graham, 2002). Für jede fehlende Beobachtung werden dabei mehrere plausible Werte generiert, indem ein zugrunde liegendes statistisches Modell angewendet wird, das häufig auf Regressionsmodellen oder bayesianischen Verfahren basiert (Little & Rubin, 2019). Daraus entstehen mehrere sogenannte Imputationsdatensätze, in denen die fehlenden Werte jeweils unterschiedlich, jedoch plausibel ergänzt werden (Little & Rubin, 2019). Diese Datensätze werden einzeln analysiert, bspw. durch die Schätzung von Mittelwerten oder Regressionskoeffizienten (Schafer & Graham, 2002). Die Ergebnisse dieser Analysen werden anschließend zu einem Gesamtergebnis zusammengeführt (Little & Rubin, 2019). Dabei werden sowohl die Unterscheide zwischen den einzelnen Imputationen als auch die Unsicherheit innerhalb der einzelnen Imputationsdatensätze berücksichtigt (Little & Rubin, 2019). Dieser Schritt ermöglicht realistische Schätzungen der Parameter und ihrer Standardfehler, da die Imputationsunsicherheit einbezogen wird (Schafer & Graham, 2002). Die Multiple Imputation ist geeignet bei einem nicht zu hohen Anteil fehlender Werte und wird in der Praxis vielfach eingesetzt, insbesondere bei MAR-Daten (Schafer & Graham, 2002). Abbildung 3.4 zeigt basierend auf Schafer und Graham (2002) vereinfacht den Ablauf der Multiple Imputation.

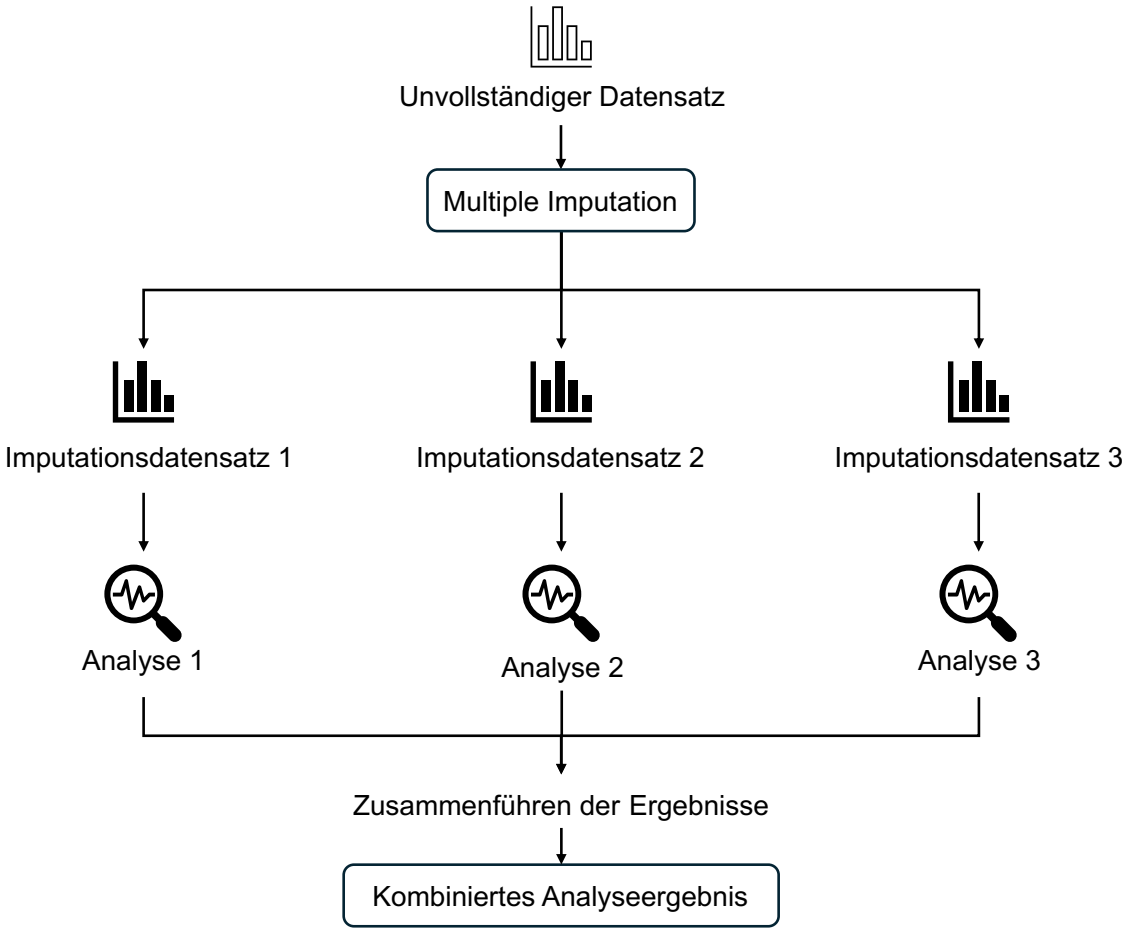


Abbildung 3.4: Vereinfachtes Ablaufdiagramm der Multiple Imputation

Das in Abbildung 3.4 dargestellte Ablaufdiagramm veranschaulicht den Prozess der Multiple Imputation. Ausgangspunkt ist ein unvollständiger Datensatz, aus dem mithilfe des Verfahrens mehrere vollständig imputierte Datensätze erzeugt werden. Jeder dieser Imputationsdatensätze wird anschließend separat analysiert, sodass mehrere Teilergebnisse vorliegen. Im letzten Schritt erfolgt ein Zusammenführen der Resultate, wodurch ein kombiniertes Analyseergebnis entsteht.

Eine weit verbreitete praktische Umsetzung der Multiple Imputation ist das Verfahren Multiple Imputation by Chained Equations (MICE), bei dem die fehlenden Werte für jede unvollständige Variable nacheinander durch bedingte Modelle auf Basis der übrigen Variablen ersetzt werden (van Buuren, 2018). Dieser Prozess läuft iterativ ab und wird mehrfach wiederholt, bis sich stabile Ergebnisse einstellen (van Buuren, 2018). In der praktischen Anwendung, etwa in der Python-Bibliothek `scikit-learn`, wird das Verfahren unter der Bezeichnung ist dieses Verfahren unter der Bezeichnung `IterativeImputer` implementiert und entspricht inhaltlich dem MICE-Verfahren (Pedregosa et al., 2011). Das Verfahren wird daher nachfolgend als iterative Imputation bezeichnet.

Expectation-Maximization-Algorithmus

Der Expectation-Maximization-Algorithmus (EM-Algorithmus) ist ebenfalls ein Verfahren, das für die Berechnung möglichst realistischer Schätzwerte für fehlende Werte aus unvollständigen Datensätzen verwendet wird (Little & Rubin, 2019). Der EM-Algorithmus basiert auf einem iterativen, zweistufigen Prozess (Little & Rubin, 2019). Im Erwartungsschritt (E-Step) werden auf Grundlage der beobachteten Daten und der aktuellen Parameterschätzungen die erwarteten Werte für die fehlenden Daten berechnet (Little & Rubin, 2019). Dabei werden beobachtete Zusammenhänge im Datensatz genutzt, um für jede unvollständige Beobachtung die fehlenden Werte anhand bedingter Wahrscheinlichkeitsverteilungen zu berechnen (Little & Rubin, 2019). Grundlage dafür sind die beobachteten Zusammenhänge zwischen den vorhandenen Variablen, aus denen unter der Annahme einer multivariaten Normalverteilung sogenannte bedingte Mittelwerte und Varianzen abgeleitet werden (Little & Rubin, 2019). Diese bedingten Parameter werden anschließend genutzt, um im Maximierungsschritt (M-Step) die Schätzwerte der Modellparameter zu optimieren (Little & Rubin, 2019). Dazu wird die sogenannte vollständige Likelihood-Funktion gebildet. Sie beschreibt die gemeinsame Wahrscheinlichkeit der beobachteten und der im E-Step geschätzten Daten (Little & Rubin, 2019). Diese Funktion wird hinsichtlich der Modellparameter maximiert, um möglichst passende Werte zu bestimmen (Little & Rubin, 2019). Beide Schritte wechseln sich ab, bis sich die Ergebnisse nicht mehr wesentlich verändern, also konvergieren (Little & Rubin, 2019).

Der EM-Algorithmus setzt voraus, dass die Verteilung der Daten entweder bekannt ist oder durch ein geeignetes Modell angenähert werden kann (Little & Rubin, 2019). Er wird typischerweise bei kontinuierlichen Variablen eingesetzt, da für diese häufig die Annahme einer multivariaten Normalverteilung als gerechtfertigt gilt (Little & Rubin, 2019). Diese Annahme ermöglicht eine mathematisch exakte Berechnung der bedingten Mittelwerte und Varianzen im E-Step (Little & Rubin, 2019). Eine Schwäche des EM-Algorithmus wiederum besteht darin, dass er die durch die Imputation entstehende Unsicherheit nicht berücksichtigt (Little & Rubin, 2019). Da nur ein einzelner Satz an Schätzwerten generiert wird, können Standardfehler und Konfidenzintervalle zu optimistisch ausfallen (Little & Rubin, 2019).

Regressionsbasierte Imputation

Bei der regressionsbasierten Imputation werden fehlende Werte durch Schätzwerte ersetzt, die mithilfe eines Regressionsmodells auf Grundlage der beobachteten Variablen ermittelt werden (Schafer & Graham, 2002). Das Verfahren nimmt an, dass sich die Zielvariable (Variable mit fehlenden Werten) aus einer oder mehreren Prädiktorvariablen vorhersagen lässt (Schafer & Graham, 2002). Als Prädiktorvariablen werden vollständig beobachtete Merkmale bezeichnet, die zur Erstellung des Regressionsmodells herangezogen werden, um die Zielgröße vorherzusagen (Little & Rubin, 2019). Bei der Erstellung des Modells werden ausschließlich Beobachtungen berücksichtigt, für die alle relevanten Prädiktorvariablen vollständig vorliegen (Little & Rubin, 2019). Anschließend werden die fehlenden Werte im ursprünglichen Datensatz durch Vorhersagen aus diesem Modell ersetzt (Little & Rubin, 2019). Die Schätzung kann deterministisch erfolgen, wobei jeweils nur der Regressionswert eingesetzt wird (Little & Rubin, 2019), oder stochastisch, indem dem Wert ein Zufallsterm entsprechend der Restvarianz des Modells hinzugefügt wird (Schafer & Graham, 2002). Letzteres erhält die natürliche Streuung der Daten besser und vermeidet eine künstliche Glättung der Verteilung

(Schafer & Graham, 2002). Die Anwendung der regressionsbasierten Imputation ist insbesondere bei kontinuierlichen Zielvariablen sinnvoll, wenn ein klarer Zusammenhang mit den Prädiktorvariablen besteht (Schafer & Graham, 2002). Die Anwendbarkeit ist jedoch eingeschränkt, wenn der Zusammenhang zwischen der Zielvariablen und den Prädiktoren nicht annähernd linear ist (Schafer & Graham, 2002) oder wenn die Prädiktorvariablen selbst fehlende Werte aufweisen (Little & Rubin, 2019). Abbildung 3.5 zeigt das Prinzip der regressionsbasierten Imputation anhand eines Diagramms.

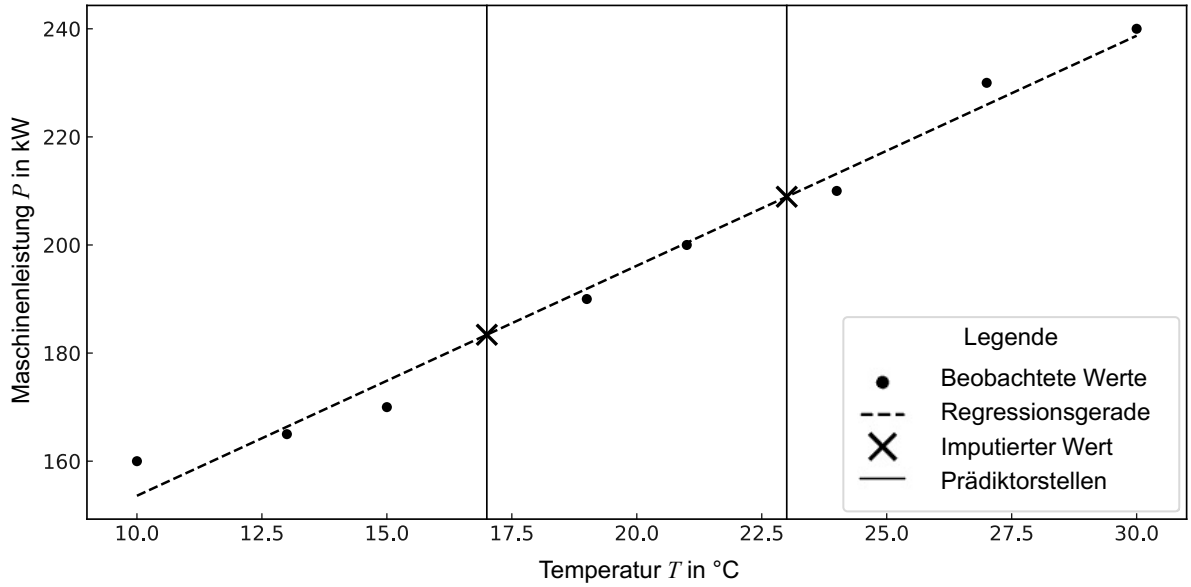


Abbildung 3.5: Prinzip der regressionsbasierten Imputation (deterministisch)

Abbildung 3.5 verdeutlicht, wie auf Grundlage eines bekanntes Zusammenhangs zwischen einer Prädiktorvariable (Temperatur) und einer Zielvariable (Maschinenleistung) fehlende Werte durch Schätzung ersetzt werden. Für vollständige Beobachtungen wird ein Regressionsmodell erzeugt, auf dessen Grundlage die Zielwerte an den bekannten Prädiktorstellen (vertikale Linien) imputiert werden. In der Abbildung erfolgt die Schätzung der Imputationswerte deterministisch. Im Fall einer stochastischen Schätzung lägen die Werte nicht genau auf der Regressionsgeraden, sondern leicht versetzt.

3.3.6 Imputation mit maschinellen Lernverfahren

Maschinelle Lernverfahren gelten als flexible und datengetriebene Ansätze, die insbesondere dann Vorteile bieten, wenn Modellannahmen wie die Normalverteilung nicht erfüllt sind (Emmanuel et al., 2021). Sie haben sich vor allem in der Datenvorverarbeitung als leistungsfähige Werkzeuge etabliert und gewinnen aufgrund ihrer vielseitigen Einsatzmöglichkeiten zunehmend an Bedeutung (Emmanuel et al., 2021). Maschinelle Lernverfahren charakterisieren sich durch ihre Fähigkeit, komplexe und nichtlineare Zusammenhänge zwischen Variablen zu erfassen und zur Rekonstruktion fehlender Daten zu nutzen (Wang et al., 2021). Im Gegensatz zu statistischen Verfahren setzen sie keine strikten Annahmen über die Verteilungsform der Daten voraus und bieten oft eine größere Flexibilität in heterogenen und hochdimensionalen Datensätzen (Prakash et al., 2024). Einige relevante Verfahren werden nachfolgend beschrieben.

k-Nearest-Neighbor-Imputation

Ein weit verbreitetes Verfahren ist die kNN-Imputation, bei der ein fehlender Wert durch eine Aggregation (z. B. Mittelwert oder Modus) der Werte der *k* ähnlichsten Beobachtungen ersetzt wird (Emmanuel et al., 2021). Dabei werden Werte von Beobachtungen zur Schätzung herangezogen, die der unvollständigen Beobachtung in ihren übrigen vorhanden Merkmalen beson-

ders ähnlich sind (Emmanuel et al., 2021). Die Anzahl *k* gibt dabei an, wie viele dieser ähnlichsten Fälle in die Berechnung mit einbezogen werden (Emmanuel et al., 2021). Die Ähnlichkeit wird dabei mithilfe einer Distanzfunktion im sogenannten Merkmalsraum bestimmt, einem mehrdimensionalen Raum, in dem jede Dimension einem beobachteten Merkmal entspricht (Emmanuel et al., 2021).

Das Verfahren ist einfach zu implementieren und erzielt gute Ergebnisse bei lokal strukturierten Daten (Hastie et al., 2009). Das bedeutet, dass sich fehlende Werte vor allem dann gut schätzen lassen, wenn es im Merkmalsraum kleinere Bereiche gibt, in denen ähnliche Beobachtungen nah beieinander liegen und ähnliche Werte aufweisen (Hastie et al., 2009). In hochdimensionalen Datensätzen hingegen verlieren Distanzmaße an Trennschärfe, wodurch die Aussagekraft des Verfahrens erheblich eingeschränkt wird (Aggarwal et al., 2001). Dieser Effekt wird als „curse of dimensionality" bezeichnet (Aggarwal et al., 2001, S. 1).

Random Forest Imputation

Ein weiteres leistungsfähiges Verfahren ist die Random Forest Imputation, insbesondere in der Variante missForest (Stekhoven & Bühlmann, 2012). Dieses Verfahren basiert auf dem in Abschnitt 2.2.1 beschriebenen Random-Forest-Ansatz nach Breiman (2001). Hier wird eine Vielzahl von Entscheidungsbäumen trainiert, wobei jeder Baum auf einer zufälligen Teilstichprobe der Daten und einer zufälligen Auswahl an Prädiktoren beruht (Breiman, 2001). Im Imputationskontext wird für jede Variable mit fehlenden Werten ein solcher Random Forest erstellt, der die vorhandenen Merkmale als Prädiktoren nutzt, um die fehlenden Werte zu schätzen (Stekhoven & Bühlmann, 2012). Die Imputation erfolgt in einem iterativen Prozess, wobei zunächst Startwerte eingesetzt werden, die meist durch einfache Verfahren wie Mittelwert- oder Modusimputation bestimmt werden (Stekhoven & Bühlmann, 2012). Anschließend werden Random-Forest-Modelle trainiert und die Schätzungen solange aktualisiert, bis eine Konvergenz erreicht ist (Stekhoven & Bühlmann, 2012).

Random Forests sind in der Lage, sowohl kategoriale als auch kontinuierliche Daten ohne spezielle Verteilungsannahmen zu verarbeiten und bieten außerdem Möglichkeiten zur Einschätzung der Vorhersagegüte, etwa über Out-of-Bag-Fehler (OOB) (Stekhoven & Bühlmann, 2012). Der OOB misst die Vorhersagegenauigkeit eines Random-Forest-Modells anhand der Daten, die bei der Konstruktion einzelner Bäume nicht verwendet wurden (Stekhoven & Bühlmann, 2012). Ein niedriger OOB steht dabei für eine hohe Genauigkeit des Modells (Stekhoven & Bühlmann, 2012). Das Verfahren zeigt insbesondere bei komplexen Interaktionen und nichtlinearen Beziehungen zwischen den Variablen eine hohe Leistungsfähigkeit und ist dabei robust gegenüber verschiedener Fehlstrukturen (Stekhoven & Bühlmann, 2012). Allerdings sind Random Forests vergleichsweise rechen- und somit auch zeitaufwändig, besonders bei großen Datensätzen mit einer Vielzahl an Bäumen und Variablen (Stekhoven & Bühlmann, 2012). Außerdem sind sie aufgrund der vielen Bäume weniger gut interpretierbar als einfache Modelle wie Regression (Breiman, 2001). Bei sehr kleinen Stichproben kann zudem die Leistung eingeschränkt sein, da sich die Fähigkeit des Random Forests verringert, generalisierende Muster zu erkennen (Hastie et al., 2009).

MIDAS

Ebenfalls den maschinellen Lernverfahren zuzuordnen ist die Multiple Imputation with Denoising Autoencoders (MIDAS), die neuronale Netze zur Imputation fehlender Werte nutzt (Prakash et al., 2024). Diese Netzwerke lernen, typische Muster in den Daten zu erkennen, indem sie während des Trainings bewusst Teile der vorhandenen Daten maskieren oder verbergen und anschließend trainiert werden, diese maskierten Werte genau wiederherzustellen (Lall & Robinson, 2022). Durch dieses Vorgehen entwickelt das Modell die Fähigkeit, fehlende Werte zu schätzen, indem es das Gelernte auf die tatsächlich fehlenden Daten anwendet (Lall & Robinson, 2022). MIDAS benötigt keine spezifischen Annahmen über die Verteilung der Daten und kann komplexe, auch nichtlineare Zusammenhänge erfassen (Lall & Robinson, 2022). Um zu vermeiden, dass das Modell die Trainingsdaten zu stark auswendig lernt (Over-

fitting), verwendet es Techniken wie Dropout, bei denen während des Trainings zufällig einzelne Neuronen (Einheiten) des Netzwerkes deaktiviert werden (Lall & Robinson, 2022). Dadurch kann laut Lall und Robinson (2022) das Netzwerk nicht zu stark von einzelnen Neuronen oder Verbindungen abhängig werden. Studien von Lall und Robinson (2022) zeigen, dass MIDAS vor allem bei großen Datensätzen mit komplexen Strukturen sehr genau und effizient arbeitet. Einschränkungen bestehen wiederum bei sehr kleinen Stichproben sowie hinsichtlich der Interpretierbarkeit, da die inneren Entscheidungsprozesse neuronaler Netze komplex sind, somit nur schwer nachvollzogen werden können und daher auch als Black Box beschrieben werden (Lall & Robinson, 2022).

Neben MIDAS existieren noch andere auf Autoencodern basierende Ansätze, bei denen Netzwerke eine komprimierte Repräsentation der vollständigen Daten lernen und daraus fehlende Werte rekonstruieren (Pereira et al., 2020). Auf eine detaillierte Darstellung dieser Verfahren wird im Folgenden aufgrund des hohen Umfangs nicht weiter eingegangen. Anschließend werden die wichtigsten Erkenntnisse zum Umgang mit fehlenden Werten zusammengefasst.

3.4 Schlussfolgerungen zum Umgang mit fehlenden Werten

Das vorliegende Kapitel befasst sich mit dem Auftreten, der Erkennung und der Behandlung fehlender Werte in Datensätzen und stellt dabei verschiedene Klassifikationen und Verfahren differenziert dar. Fehlende Werte stellen ein zentrales Problem in der Datenanalyse dar, da sie die Integrität und Aussagekraft statistischer und maschineller Lernverfahren erheblich beeinträchtigen können (Little & Rubin, 2019). Fehlende Werte können unterschiedliche Ursachen haben, etwa technische Defekte (Teh et al., 2020) oder systembedingte Ausfälle in industriellen Prozessen (Fleck et al., 2020). Entscheidend für die Auswahl geeigneter Behandlungsverfahren ist die genaue Kenntnis des zugrunde liegenden Mechanismus des Fehlens von Werten (Little & Rubin, 2019). Gemäß der Klassifikation von Little und Rubin (2019) lassen sich drei Arten fehlender Werte unterscheiden. Demnach erfolgt das Fehlen von Werten bei MCAR völlig unabhängig von beobachteten und unbeobachteten Werten, wohingegen das Fehlen bei MAR durch beobachtete Variablen erklärt werden kann. Bei MNAR liegt gemäß der Einordnung der Autoren die Ursache des Fehlens in der Variable selbst. Die Identifikation fehlender Werte erfolgt primär im Rahmen einer EDA (Cleff, 2015), etwa durch grafische Verfahren wie Heatmaps (Fernstad, 2019) oder Missingness-Matrizen (Little & Rubin, 2019), welche Muster systematischen Fehlens visualisieren. Ergänzend kommen statistische Tests zum Einsatz, etwa zur Prüfung der MCAR-Annahme (Little & Rubin, 2019).

Zur Behandlung fehlender Werte existieren nach Little und Rubin (2019) unterschiedliche Arten von Verfahren. Eliminierungsverfahren bieten demnach einen einfachen Zugang, sind jedoch ausschließlich bei vollständig zufälligem Fehlen der Daten ohne gravierenden Informationsverlust anwendbar. Imputationsverfahren hingegen ermöglichen eine Erhaltung der Datenstruktur und reduzieren Verzerrungen (Schafer & Graham, 2002), wobei Aufwand und Komplexität je nach Verfahren stark variieren (Emmanuel et al., 2021). Statistische Imputationsverfahren sind unkompliziert und für eine geringe Anzahl fehlender Werte geeignet (García et al., 2015), können jedoch Varianz und Kovarianz unterschätzen (Little & Rubin, 2019). Klassenbasierte Verfahren berücksichtigen Gruppenzugehörigkeiten und optimieren dadurch die Plausibilität der Schätzungen (Bankhofer, 1995), stoßen jedoch bei komplexen multivariaten Zusammenhängen an ihre Grenzen (Schafer & Graham, 2002). Multivariate Verfahren erfassen diese Zusammenhänge umfassend und bieten eine solide statistische Grundlage, sind jedoch rechnerisch aufwändiger (Schafer & Graham, 2002). Ansätze auf Basis maschineller Lernverfahren erweitern das Spektrum durch ihre Fähigkeit, komplexe, nichtlineare Muster zu erkennen und zeigen insbesondere bei heterogenen und hochdimensionalen Daten gute Ergebnisse (Emmanuel et al., 2021). Die Wahl des Verfahrens sollte stets an den spezifischen Kontext angepasst und die Ergebnisse sorgfältig validiert werden (Emmanuel et al., 2021). Nach der Behandlung fehlender Werte ist die Reduktion überflüssiger Informationen eine weitere bedeutende Komponente der Datenvorverarbeitung (García et al., 2015). Im folgenden Kapitel werden daher Verfahren zur Erkennung und Entfernung redundanter Daten vorgestellt.

4 Umgang mit Datenredundanzen

In datenanalytischen Prozessen sind umfangreiche und hochdimensionale Datensätze häufig mit redundanten oder irrelevanten Informationen behaftet (Han et al., 2012). Solche Daten erschweren die Analyse und können die Effizienz der eingesetzten Data-Mining-Verfahren beeinträchtigen (Han et al., 2012). Die gezielte Reduktion redundanter oder irrelevanter Daten ist daher ein entscheidender Schritt in der Datenvorverarbeitung, da sie die Effizienz von Analyseverfahren verbessert und die resultierenden Modelle verständlicher macht (Han et al., 2012).

Dieses Kapitel stellt verbreitete Verfahren zur Reduktion redundanter Daten vor. Die Darstellung der Verfahren erfolgt mit dem Ziel, deren methodische Grundlagen zu vermitteln und die jeweiligen Einsatzmöglichkeiten im datenanalytischen Kontext zu erläutern. Anders als im Kapitel zum Umgang mit fehlenden Werten wird hier die Trennung zwischen Verfahren zur Identifikation und zur Behandlung bewusst aufgehoben, da viele Verfahren beide Aufgaben gleichzeitig erfüllen. Zur Einordnung der Verfahren werden im folgenden Abschnitt zunächst die verschiedenen Typen von Datenredundanzen beschrieben.

4.1 Typen von Datenredundanzen

In der Literatur wird bei der Datenreduktion zwischen Merkmalsreduktion und Instanzenreduktion unterschieden (Han et al., 2012). Redundanzen auf Merkmalsebene liegen vor, wenn zwei oder mehr Attribute einer Datenstruktur dieselbe oder nahezu dieselbe Information enthalten (Han et al., 2012). Solche Merkmale beeinträchtigen die Effizienz von Data-Mining-Verfahren und können den Rechenaufwand deutlich erhöhen (Han et al., 2012). Darüber hinaus können sie statistische und maschinelle Lernmodelle überlasten, die im Rahmen von Datenanalyseverfahren verwendet werden, da sie zu Multikollinearität führen und dadurch die Stabilität der Modellschätzung beeinträchtigen (Kim, 2019). Zudem steigt mit zunehmender Anzahl redundanter Merkmale die Komplexität des Modells, ohne dass ein entsprechender Informationsgewinn erzielt wird (Kuzudisli et al., 2023). Redundanz auf Instanzebene hingegen liegt vor, wenn Beobachtungen im Datensatz enthalten sind, deren Informationen durch andere Instanzen bereits abgedeckt sind, weil sie identische oder sehr ähnliche Merkmalsausprägungen besitzen (Olvera-López et al., 2010). Solche redundanten Instanzen vergrößern die Datenmenge, führen zu längeren Rechenzeiten und können die Modellanpassung erschweren (Garcia et al., 2012). Mithilfe geeigneter Verfahren lassen sich Datenredundanzen erkennen und reduzieren, ohne den strukturellen Informationsgehalt des Datensatzes wesentlich zu beeinträchtigen (Triguero et al., 2015). Nachdem nun die Typen von Datenredundanzen beschrieben wurden, werden im Folgenden zunächst Verfahren zur Reduktion redundanter Merkmale vorgestellt, bevor anschließend Verfahren zur Behandlung redundanter Instanzen betrachtet werden.

4.2 Verfahren zur Reduktion von Datenredundanzen auf Merkmalsebene

Um redundante Merkmale zu identifizieren und zu entfernen, stehen unterschiedliche Verfahren zur Verfügung, die je nach Art der Daten, Zielvariable und später eingesetztem Lernalgorithmus variieren können (Theng & Bhoyar, 2024). Einige Verfahren arbeiten auf Grundlage statistischer Korrelationen, andere basieren auf informationstheoretischen Konzepten oder sind direkt in den Lernprozess eingebettet (Li et al., 2017). Die Auswahl eines geeigneten Verfahrens beeinflusst nicht nur die Effizienz der Datenverarbeitung, sondern auch die Qualität und Interpretierbarkeit der resultierenden Modelle (Brown et al., 2012). Um die unterschiedlichen methodischen Ansätze zu betrachten, werden im Folgenden grundlegende Verfahren zur Reduktion redundanter Merkmale auf Merkmalsebene vorgestellt.

4.2.1 Korrelationsanalyse

Die Korrelationsanalyse zählt zu den grundlegenden Verfahren der Merkmalsauswahl und dient der Identifikation redundanter Informationen in Datensätzen (Guyon & Elisseeff, 2003). Ziel ist es, Merkmale zu erkennen, die untereinander stark korrelieren und somit potenziell dieselbe Information enthalten (Guyon & Elisseeff, 2003). Ein häufig eingesetztes Werkzeug ist die Pearson-Korrelation ρ (Guyon & Elisseeff, 2003), die den linearen Zusammenhang zwischen zwei Merkmalen quantifiziert und Werte zwischen -1 und +1 annehmen kann (Cleff, 2015). In der Praxis werden oft folgende Kategorien verwendet, um die Stärke des linearen Zusammenhangs zu interpretieren (Cleff, 2015, S. 101):

- $|r| < 0{,}5$: Schwacher linearer Zusammenhang
- $0{,}5 \leq |r| < 0{,}8$: Mittlerer linearer Zusammenhang
- $|r| \geq 0{,}8$: Großer linearer Zusammenhang

Liegt eine starke Korrelation zwischen zwei Merkmalen vor, kann dies auf redundante Information hinweisen (Li et al., 2017). In der Praxis wird in solchen Fällen häufig eines der Merkmale entfernt, um den Informationsgehalt zu verdichten und die Modellstruktur zu vereinfachen (Li et al., 2017). Hall (1999) greift diese Idee auf und erweitert sie im Rahmen des Verfahrens der Correlation-based Feature Selection (CFS) zu einem systematischen Auswahlverfahren. In seinem Ansatz der CFS schlägt Hall (1999) vor, aus der Gesamtheit aller Merkmale eine Teilmenge so auszuwählen, dass diese einerseits eine hohe Korrelation mit der Zielvariable (Relevanz) aufweist und andererseits möglichst gering untereinander korreliert (geringe Redundanz). Um dies zu erreichen, wird für jede potenzielle Merkmalskombination ein sogenannter Merit-Wert berechnet, der das Verhältnis zwischen der durchschnittlichen Korrelation der Merkmale mit der Zielvariablen (Relevanz) und der durchschnittlichen Korrelation der Merkmale untereinander (Redundanz) ausdrückt (Hall, 1999). Je höher dieser Wert, desto besser die Balance zwischen hoher Informationsdichte und geringer Redundanz (Hall, 1999). Ziel dieser Auswahl ist es, eine kompakte, aber informationsreiche Merkmalsbasis zu identifizieren, die sich für den Einsatz in nachfolgenden Analysen eignet (Hall, 1999). Korrelationsbasierte Verfahren ermöglichen somit eine effektive Vorselektion von Merkmalen und dienen als Grundlage für weiterführende Reduktionsverfahren (Li et al., 2017). Besonders im Kontext hochdimensionaler Daten können sie dazu beitragen, die Komplexität zu verringern, die Modellgüte zu steigern und Overfitting zu vermeiden (Li et al., 2017).

4.2.2 Mutual-Information-Verfahren

Mutual Information (MI) ist ein informations-theoretisches Maß, das den gemeinsamen Informationsgehalt zweier Variablen quantifiziert (Cover & Thomas, 2005). MI basiert auf dem Konzept der Entropie und misst, wie stark das Wissen über den Wert eines Merkmals die Unsicherheit über ein anderes Merkmal reduziert (Cover & Thomas, 2005). Je größer dieser Informationsgewinn, desto stärker ist die wechselseitige Abhängigkeit zwischen beiden Merkmalen (Cover & Thomas, 2005). Dabei ist MI symmetrisch, nicht negativ und nimmt den Wert null an, wenn zwei Merkmale vollkommen unabhängig voneinander sind (Cover & Thomas, 2005).

Im Gegensatz zur Pearson-Korrelation ist MI nicht auf lineare Zusammenhänge beschränkt, sondern erkennt auch nichtlineare Zusammenhänge (Cover & Thomas, 2005). Aufgrund dieser Eigenschaft wird MI wird in zahlreichen Verfahren der Merkmalsauswahl (Feature Selection) genutzt, um Redundanzen zwischen Merkmalen zu identifizieren (Vergara & Estévez, 2014). Hohe MI-Werte zwischen zwei Features gelten dabei als Hinweis auf ähnliche Informationsinhalte. Verfahren wie Minimum Redundancy Maximum Relevance folgen dieser Einsicht, indem sie Merkmale mit hoher Relevanz zur Zielvariable und gleichzeitig geringer Redundanz untereinander selektieren (Radovic et al., 2017). Aufgrund ihrer Flexibilität eignet sich MI insbesondere zur Analyse hochdimensionaler und nichtlinear strukturierter Datensätze (Beraha et al., 2019).

4.2.3 Variance-Inflation-Factor-Verfahren

Der Variance Inflation Factor (VIF) ist ein etabliertes Maß zur Diagnose von Multikollinearität in linearen Regressionsmodellen (Kyriazos & Poga, 2023). Er beschreibt, in welchem Ausmaß die Varianz einzelner Regressionskoeffizienten durch lineare Zusammenhänge mit anderen Prädiktoren vergrößert wird (James et al., 2023). Ein hoher VIF deutet darauf hin, dass das betreffende Merkmal im Modell keine oder kaum eigenständige Information mehr beiträgt (Akinwande et al., 2015). Der VIF wird insbesondere eingesetzt, um redundante Merkmale im Rahmen linearer Modellierungen zu identifizieren (Akinwande et al., 2015).

Das grundlegende Prinzip des VIF besteht darin, zu prüfen, in welchem Maße sich ein Merkmal durch andere erklärende Variablen vorhersagen lässt (Akinwande et al., 2015). Je stärker ein Merkmal durch andere Merkmale erklärbar ist, desto wahrscheinlicher handelt es sich um eine potenziell redundante Information (O'Brien, 2007). Der VIF wird für jede Variable berechnet, indem untersucht wird, wie gut sie durch die übrigen Prädiktoren erklärbar ist, typischerweise unter Rückgriff auf das Bestimmtheitsmaß (James et al., 2023). Der Wert eins stellt das theoretische Minimum des VIF dar und signalisiert, dass keinerlei lineare Beziehung zwischen Variablen besteht (James et al., 2023). Als Faustregel wird häufig angenommen, dass ein VIF-Wert größer als fünf oder zehn auf problematische Multikollinearität hinweist (James et al., 2023). Hohe VIF-Werte können die Interpretierbarkeit und Stabilität eines Modells beeinträchtigen, da sie auf starke lineare Zusammenhänge zwischen den Prädiktoren hinweisen (James et al., 2023). Der VIF gilt aufgrund seiner Fähigkeit, die lineare Erklärbarkeit eines Merkmals durch andere Prädiktoren zu quantifizieren, als bewährtes Werkzeug zur Erkennung redundanter Merkmale in linearen Modellkontexten (Akinwande et al., 2015).

4.2.4 Wrapper-Verfahren

Wrapper-Verfahren bewerten die Bedeutung einzelner Merkmale, indem sie deren Einfluss auf die Leistung eines spezifischen Lernmodells direkt testen (Kohavi & John, 1997). Dazu wird ein Lernalgorithmus, bspw. ein Entscheidungsbaum oder Klassifikator, wiederholt auf unterschiedliche Teilmengen der Merkmale trainiert (Guyon & Elisseeff, 2003). Für jede Merkmalskombination wird die Modellgüte berechnet, um herauszufinden, welche Merkmale zur besten Vorhersage beitragen (Bolón-Canedo et al., 2013). Die Modellgüte bezeichnet dabei die Fähigkeit des Modells, korrekte Vorhersagen für unbekannte Daten zu liefern (Kohavi & John, 1997). Dies gelingt etwa durch eine hohe Genauigkeit oder geringe Fehlermaße (Kohavi & John, 1997).

Ein verbreitetes Wrapper-Verfahren ist die Recursive Feature Elimination (RFE), bei der das Modell mit allen verfügbaren Merkmalen gestartet wird (Guyon et al., 2002). Anschließend wird iterativ das Merkmal mit dem geringsten Einfluss auf die Modellgüte entfernt, bis eine ökonomische Merkmalsmenge gefunden ist (Guyon et al., 2002). Auf diese Weise lässt sich eine Merkmalsauswahl erzeugen, die eng an die Leistung des konkreten Modells angepasst ist (Kohavi & John, 1997). Wrapper-Verfahren gelten daher als besonders genau, sind aber rechenintensiv, da für jede Merkmalskonstellation ein vollständiges Modell trainiert werden muss (Liu & Yu, 2005). Außerdem arbeiten sie stets in Abhängigkeit eines spezifischen Lernalgorithmus und sind somit zwar auf das verwendete Modell abgestimmt, jedoch nicht notwendigerweise auf andere Modelltypen übertragbar (Blum & Langley, 1997).

4.2.5 Embedded-Verfahren

Embedded-Verfahren verknüpfen die Merkmalsauswahl unmittelbar mit dem Lernprozess eines Modells (Saeys et al., 2007). Im Gegensatz zu Wrapper-Verfahren, bei denen die Bewertung der Merkmale separat durch wiederholtes Modelltraining erfolgt, ist die Auswahl hier ein integraler Bestandteil des Trainingsvorgangs selbst (Chandrashekar & Sahin, 2014). Das bedeutet, dass das Modell bereits während des Trainings erkennt, welche Merkmale für die Zielgröße relevant sind und welche nicht (Chandrashekar & Sahin, 2014). Ein Beispiel für Embedded-Verfahren ist die Lasso-Regression (Tibshirani, 1996). Dieses Verfahren regt das Modell dazu an, möglichst viele Merkmalsgewichte auf null zu setzen, wodurch irrelevante Merkmale

automatisch ausgeschlossen werden (Tibshirani, 1996). Das bedeutet, dass Merkmale, die wenig zur Vorhersage beitragen, automatisch ausgeschlossen werden (Tibshirani, 1996). Auf diese Weise wird während des Modelltrainings gleichzeitig eine Auswahl relevanter Merkmale getroffen (Tibshirani, 1996). Auch Entscheidungsbäume wie Random Forests verfügen über eingebettete Selektionsmechanismen, indem sie beim Aufbau des Modells wiederholt Splits durchführen, also Entscheidungen darüber treffen, bei welchem Schwellenwert eines Merkmals der Datenraum aufgeteilt wird (Breiman, 2001). Je häufiger ein Merkmal für solche Aufteilungen verwendet wird, desto größer ist sein Einfluss auf die Modellstruktur, was Rückschlüsse auf seine Relevanz zulässt (Breiman, 2001).

Embedded-Verfahren führen die Merkmalsauswahl während des Trainings durch und gelten als Kompromiss zwischen rechenintensiven Wrapper-Verfahren und modellunabhängigen Filterverfahren (Chandrashekar & Sahin, 2014). Sie sind meist effizienter als Wrapper-Verfahren, da keine separate Suche über alle Merkmalskombinationen erforderlich ist (Liu & Motoda, 2007). Gleichzeitig berücksichtigen sie im Gegensatz zu Filter-Verfahren auch die Interaktion mit dem verwendeten Lernalgorithmus (Liu & Motoda, 2007). Allerdings sind auch Embedded-Verfahren modellabhängig, die ausgewählten Merkmale sind also auf das jeweilige Modell angepasst und nicht notwendigerweise übertragbar (Liu & Motoda, 2007).

4.2.6 Merkmalsextraktionsverfahren

Im Gegensatz zur Merkmalsauswahl, bei der eine Teilmenge bestehender Merkmale beibehalten wird, zielt die Merkmalsextraktion darauf ab, neue Merkmale durch Transformation der ursprünglichen Daten zu erzeugen (Li et al., 2017). Diese neuen Merkmale stellen Kombinationen der ursprünglichen Variablen dar und repräsentieren die relevanten Strukturen des Datensatzes in kompakterer Form (Li et al., 2017). Ziel ist es, die Anzahl der Merkmale zu verringern und gleichzeitig die relevanten Strukturen der Daten zu erhalten (Liu & Motoda, 2007).

Ein verbreitetes Verfahren ist die PCA, bei der die Daten in ein neues orthogonales Koordinatensystem transformiert werden, dessen Achsen (Hauptkomponenten) die maximale Varianz der Daten abbilden (Neuer, 2024). Neben linearen Verfahren wie PCA existieren auch nichtlineare Ansätze wie t-Distributed-Stochastic-Neighbor-Embedding-Algorithm die besonders für die Visualisierung hochdimensionaler Daten geeignet sind, da sie lokale Nachbarschaften und Clusterstrukturen abbilden können (Neuer, 2024). Die Merkmalsextraktion dienen somit nicht nur der Effizienzsteigerung (Li et al., 2017), sondern kann auch zur Entdeckung verborgener Muster und Strukturen beitragen, die im ursprünglichen Merkmalsraum nicht unmittelbar sichtbar sind (Neuer, 2024). Damit leistet die Merkmalsextraktion einen wichtigen Beitrag zur Aufbereitung komplexer Daten. Nach der Vorstellung der Verfahren zur Merkmalsreduktion werden im folgenden Abschnitt Verfahren zur Reduktion auf Instanzebene beschrieben.

4.3 Verfahren zur Reduktion von Datenredundanzen auf Instanzebene

Zur Bestimmung der Ähnlichkeit zwischen Instanzen werden laut Wilson und Martinez (2000) in der Regel Distanz- oder Ähnlichkeitsmaße verwendet. Instanzen, die im Merkmalsraum in unmittelbarer Nähe zueinander liegen und identische Zielvariablen aufweisen, gelten demnach häufig als potenziell redundant, da sie inhaltlich keine zusätzlichen Informationen zum Datensatz beitragen (Wilson & Martinez, 2000). Redundante Instanzen können gelöscht, zusammengefasst oder durch sogenannte Prototypen ersetzt werden (Wilson & Martinez, 2000). Die Wahl des konkreten Verfahrens hängt von Anwendungsfall, der Datenstruktur und dem Ziel der Analyse ab (Triguero et al., 2015). Bei begrenzten Speicher- oder Rechenressourcen kann das Entfernen von Duplikaten oder sehr ähnlichen Instanzen eine einfache und effektive Maßnahme zur Reduktion des Speicherbedarfs sein (Wilson & Martinez, 2000). In anderen Szenarien, etwa bei der Modellkomprimierung oder bei Anwendungen mit kontinuierlich einlaufenden Datenströmen, sind weiterführende Selektionsstrategien wie das Clustering oder Prototyping zielführender (Triguero et al., 2015). Im Folgenden werden relevante Verfahren zur Identifikation und Reduktion redundanter Instanzen vorgestellt.

4.3.1 Distanzbasierte Analyse

Ein verbreitetes Verfahren zur Erkennung redundanter Instanzen ist die distanzbasierte Analyse (Aggarwal et al., 2001). Dabei werden für jede Instanz Abstände zu anderen Instanzen im Merkmalsraum berechnet, etwa mithilfe der euklidischen oder Manhattan-Distanz (Aggarwal et al., 2001). Die euklidische Distanz entspricht der Luftlinienentfernung zwischen zwei Datenpunkten im Merkmalsraum und berücksichtigt alle Dimensionen gleichermaßen (Aggarwal et al., 2001). Die Manhattan-Distanz hingegen summiert die absoluten Unterschiede in jeder Dimension und ist dadurch besonders robust gegenüber Ausreißern (Aggarwal et al., 2001). Beide Maße ermöglichen die Quantifizierung von Ähnlichkeiten zwischen Instanzen, wobei geringe Distanzen auf potenzielle Redundanz hindeuten können (Aggarwal et al., 2001). Basierend auf einem zuvor festgelegten Distanzschwellenwert können solche Instanzen identifiziert und als redundant klassifiziert werden (Wilson & Martinez, 2000). Diese werden dann entweder entfernt, zusammengefasst oder durch Repräsentanten ersetzt (Wilson & Martinez, 2000).

Distanzbasierte Verfahren werden häufig zur Datenbereinigung eingesetzt, etwa zur Erkennung und Entfernung duplizierter oder sehr ähnlicher Instanzen, die als redundant gelten, da sie keine zusätzlichen Informationen zur Struktur des Datensatzes beitragen (Wilson & Martinez, 2000). Sie lassen sich zudem gut mit Clustering- oder Prototypenverfahren kombinieren (Triguero et al., 2015). In hochdimensionalen Datensätzen kann jedoch die Aussagekraft von Distanzmaßen abnehmen (Aggarwal et al., 2001). Mit zunehmender Anzahl an Merkmalen nähern sich die Abstände zwischen allen Datenpunkten an, wodurch sich ihre Unterscheidbarkeit reduziert, weshalb auf Distanzmaßen basierende Verfahren fehleranfällig oder ineffizient werden können (Aggarwal et al., 2001). Um dem entgegenzuwirken, empfiehlt sich der Einsatz dimensionsreduzierender DVV, wie bspw. der Hauptkomponentenanalyse (Gewers et al., 2022). Basierend auf Aggarwal et al. (2001) zeigt die folgende Abbildung 4.1 schematisch den Unterschied der euklidischen und der Manhattan-Distanz.

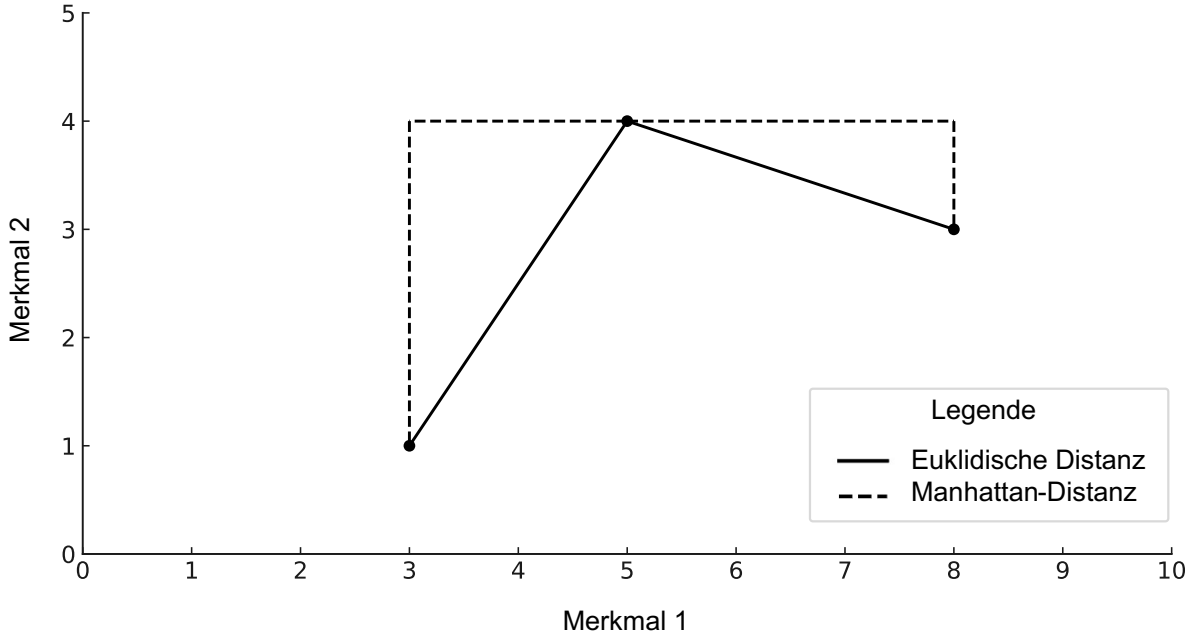


Abbildung 4.1: Schematischer Vergleich der euklidischen und der Manhattan-Distanz

Abbildung 4.1 veranschaulicht den Unterscheid zwischen der euklidischen Distanz und der Manhattan-Distanz anhand zweier Punkte im Merkmalsraum. Während die euklidische Distanz der direkten Luftlinie zwischen den Punkten entspricht, basiert die Manhattan-Distanz auf einem rechtwinkligen Pfad, der nur horizontale und vertikale Schritte erlaubt. Die Darstellung zeigt, dass sich die beiden Distanzmaße je nach Anordnung deutlich unterscheiden können.

4.3.2 Clustering

Ein weiterer verbreiteter Ansatz zur Erkennung von redundanten Informationen auf Instanzebene ist das Clustering (Olvera-López et al., 2010). Hierbei werden Instanzen basierend auf ihrer Ähnlichkeit in Cluster eingeteilt, sodass innerhalb eines Clusters eine hohe Ähnlichkeit und zwischen Clustern eine möglichst geringe Ähnlichkeit besteht (Aggarwal, 2018). Verfahren wie der k-Means-Algorithmus erlauben es, innerhalb dieser Cluster sogenannte Repräsentanten oder Prototypen zu identifizieren (Aggarwal, 2018). Beim k-Means-Algorithmus wird für jedes Cluster ein Zentrum berechnet, das den Durchschnitt aller zugehörigen Punkte darstellt (Aggarwal, 2018). Dieses Zentrum wird auch als Prototyp bezeichnet und kann als typischer Vertreter für alle ähnlichen Instanzen innerhalb des entsprechenden Clusters dienen (Aggarwal, 2018). Instanzen, die sich nur geringfügig von diesen Prototypen unterscheiden, tragen häufig keine zusätzlichen Informationen bei (Aggarwal, 2018) und können daher entfernt oder aggregiert werden.

4.3.3 Instanzenselektion

Neben distanzbasierten Verfahren und Clustering stellt die gezielte Instanzenselektion ein weiteres grundlegendes Vorgehen zur Reduktion redundanter Daten auf Instanzebene dar (Olvera-López et al., 2010). Ziel ist dabei, eine Teilmenge der Instanzen zu extrahieren, die sowohl repräsentativ für die Gesamtstruktur ist, als auch zur Erhöhung der Modellgüte beiträgt (Triguero et al., 2015). Die Verfahren der Instanzenselektion lassen sich grob unterteilen in konsistenzorientierte und kompressionsorientierte Ansätze, je nachdem, ob Genauigkeit oder Datenreduktion im Vordergrund steht (Wilson & Martinez, 2000). Ein Beispiel für einen konsistenzorientierten Ansatz ist das Edited Nearest Neighbor (ENN)-Verfahren, bei dem inkonsistente oder potenziell fehlerhafte Instanzen entfernt werden, die von ihren Nachbarn anders klassifiziert würden (Triguero et al., 2015). Demgegenüber steht das Verfahren Condensed Nearest Neighbor, das als kompressionsorientierter Ansatz nur jene Instanzen beibehält, die für eine korrekte Klassifikation tatsächlich notwendig sind (Wilson & Martinez, 2000). Viele dieser Verfahren wurden ursprünglich im Kontext des überwachten Lernens entwickelt und zielen auf eine Verbesserung der Klassifikationsleistung bei gleichzeitiger Reduktion der Trainingsdatenmenge ab (Wilson & Martinez, 2000). Sie lassen sich jedoch auch adaptieren, um in nichtüberwachten Analysekontexten redundante Instanzen zu identifizieren (Olvera-López et al., 2010).

4.4 Schlussfolgerungen zum Umgang mit Datenredundanzen

Die Reduktion redundanter Informationen ist ein grundlegender Bestandteil der Datenvorverarbeitung, da überflüssige Merkmale oder Instanzen die Modellqualität, Rechenleistung und Interpretierbarkeit negativ beeinflussen können (Han et al., 2012). Besonders bei hochdimensionalen Datensätzen treten solche Redundanzen häufig auf und müssen systematisch adressiert werden (Kim, 2019). Zur Reduktion redundanter Merkmale kommen insbesondere korrelationsbasierte Verfahren wie CFS zum Einsatz, die auf eine geringe Redundanz bei gleichzeitig hoher Zielrelevanz abzielen (Hall, 1999). Informationsbasierte Ansätze wie MI erlauben zusätzlich die Erkennung nichtlinearer Zusammenhänge (Cover & Thomas, 2005), während der VIF speziell zur Diagnose von Multikollinearität in linearen Modellen dient (Akinwande et al., 2015). Wrapper- und Embedded-Verfahren bieten modellabhängige Alternativen, die entweder durch wiederholtes Training (Guyon et al., 2002) oder direkt während des Modellaufbaus (Tibshirani, 1996) relevante Merkmale selektieren. Auch auf Instanzebene ist die Entfernung redundanter Daten von Bedeutung. Distanzbasierte Verfahren können sehr ähnliche Beobachtungen identifizieren (Aggarwal et al., 2001). Clustering-Verfahren wiederum ermöglichen die Bildung typischer Repräsentanten (Aggarwal, 2018), während Instanzenselektionsverfahren gezielt besonders informative Teilmengen extrahieren (Wilson & Martinez, 2000). Auf Basis der in den Kapiteln 3.3, 4.2 und 4.3 vorgestellten Verfahren zur Behandlung fehlender Werte und zur Datenreduktion wird im folgenden Kapitel ein Experimentierplan entwickelt, anhand dessen dem die Wirksamkeit der Verfahren systematisch überprüft werden kann.

5 Konzeption des Experimentierplans

Um den Einfluss verschiedener DVV auf datengetriebene Analyseprozesse nachvollziehbar zu untersuchen, ist ein methodisch fundiertes Vorgehen erforderlich. Im Zentrum steht dabei die Entwicklung eines strukturierten, transparenten und universell anwendbaren Experimentierplans, mit dem sich der Einfluss verschiedener Strategien zur Behandlung fehlender Werte und zur Reduktion von Datenredundanzen auf die Qualität datengetriebener Analyseprozesse messen lässt. Dieser Plan bildet die methodische Grundlage für die spätere Implementierung und soll eine objektive Vergleichbarkeit unterschiedlicher Verfahren unter reproduzierbaren Bedingungen ermöglichen.

Ein solches Vorgehen ist notwendig, weil sich in der Praxis zeigt, dass die Leistungsfähigkeit datengetriebener Modelle, wie in Kapitel 2 dargelegt, nicht allein vom gewählten Analyseverfahren abhängt, sondern in erheblichem Maße von der Qualität und Struktur der Eingangsdaten beeinflusst wird. Fehlende Werte, redundante oder irrelevante Daten stellen grundlegende Herausforderungen dar, die im Rahmen der Datenvorverarbeitung adressiert werden müssen. Gleichzeitig existieren in der Literatur und Praxis keine einheitlichen Empfehlungen zur Auswahl oder Kombination geeigneter Verfahren. Vielmehr sind die Ergebnisse stark kontextabhängig und variieren je nach Art, Umfang und Struktur des zugrunde liegenden Datensatzes. Um diesem Umstand methodisch zu begegnen, ist ein flexibles und zugleich systematisches Versuchsdesign erforderlich, das verschiedene Vorverarbeitungsstrategien unter identischen Rahmenbedingungen vergleichbar macht.

Für die Entwicklung eines belastbaren Experimentierplans ist es notwendig, geeignete Kriterien zur Auswahl von Evaluationsmetriken zu definieren. Diese Metriken sollen es ermöglichen, die Auswirkungen unterschiedlicher DVV auf datengetriebene Analyseprozesse systematisch und vergleichbar zu bewerten. Dabei stehen Bewertungsmaßstäbe im Vordergrund, die eine zuverlässige Unterscheidung zwischen geeigneten und weniger geeigneten Verfahren erlauben, ohne dabei stark vom eingesetzten Modell oder spezifischen Datensätzen abhängig zu sein. Relevante Auswahlkriterien umfassen unter anderem die Aussagekraft hinsichtlich Analysequalität, die Übertragbarkeit auf verschiedene Datenstrukturen, sowie die Verbreitung und wissenschaftliche Anerkennung in der Praxis. Ergänzend sind auch Effizienzmetriken zu berücksichtigen, etwa ein Bewertungsmaß, welches praxisnahe Aussagen über den Ressourcenbedarf eines Verfahrens zulässt. Schließlich ist es erforderlich, die Stabilität von Ergebnissen gegenüber Veränderungen in der Datenbasis zu erfassen, weshalb auch Kriterien zur Bewertung der Robustheit einbezogen werden.

Neben der Auswahl der Metriken ist zu definieren, welche Anforderungen an die Verfahren selbst zu stellen sind, um in den Experimentierplan aufgenommen zu werden. Die Festlegung solcher Kriterien gewährleistet, dass die Auswahl nachvollziehbar, theoriegestützt und praktisch umsetzbar ist. Zu den zentralen Aspekten zählen die Verfügbarkeit und Anwendbarkeit der Verfahren mit gängigen Softwarebibliotheken, ihre Unabhängigkeit von spezifischen Modellarchitekturen sowie ihre inhaltliche Relevanz für reale datenbasierte Fragestellungen. Darüber hinaus sollten die Verfahren eine methodische Vielfalt abdecken, um sowohl einfache als auch komplexere Ansätze in die Analyse einbeziehen zu können. Die tatsächliche Auswahl der Verfahren erfolgt darauf aufbauend in der Implementierung.

Den Kern des methodischen Vorgehens bildet der Aufbau des eigentlichen Experimentierplans. Dieser sieht vor, dass sowohl einzelne DVV als auch deren Kombinationen untersucht werden. Darüber hinaus wird die Reihenfolge der Anwendung variiert, um zu prüfen, inwiefern diese einen Einfluss auf die Analysequalität hat. Ziel ist es, Wechselwirkungen und Kombinationseffekte zwischen Verfahren identifizierbar zu machen und daraus Empfehlungen für den methodischen Aufbau datenbasierter Analysen abzuleiten. Das Design wird so konzipiert, dass es grundsätzlich auf unterschiedliche Datensätze übertragbar ist und damit nicht nur exemplarischen, sondern auch generalisierbaren Erkenntnisgewinn ermöglicht.

Die Implementierung dieses Plans erfolgt im nachfolgenden Kapitel, in dem die experimentellen Schritte anhand eines geeigneten Beispieldatensatzes durchgeführt und ausgewertet werden. Die hier beschriebene Methodik stellt die konzeptionelle Grundlage für eine systematische und nachvollziehbare Bewertung datenbezogener Vorverarbeitungsstrategien dar. Um dabei belastbare und vergleichbare Ergebnisse zu erzielen, ist im ersten Schritt die Festlegung geeigneter Evaluationsmetriken erforderlich, anhand derer sich die Auswirkungen einzelner Maßnahmen quantifizieren lassen.

5.1 Kriterien zur Auswahl geeigneter Evaluationsmetriken

Die Evaluierung datenbasierter Analyseverfahren setzt, wie in Abschnitt 2.2.2 beschrieben, eine fundierte Auswahl geeigneter Metriken voraus. Insbesondere bei der Bewertung von DVV, deren Effekte nicht unmittelbar sichtbar sind, sondern sich indirekt über nachgelagerte Data-Mining-Verfahren ausdrücken, ist eine strukturierte Auswahl notwendig. Ziel dieses Kapitels ist es, die methodischen, theoretischen und praktischen Grundlagen für die Wahl geeigneter Evaluationsmetriken zu entwickeln. Diese Metriken ermöglichen den systematischen Vergleich verschiedener DVV und deren Kombinationen im Rahmen eines experimentellen Designs.

5.1.1 Ziel und Funktion von Evaluationsmetriken

Evaluationsmetriken übernehmen in dieser Arbeit eine grundlegende methodische Funktion für die vergleichende Bewertung von Verfahren zur Behandlung fehlender Werte und zur Reduktion redundanter Merkmale und Instanzen. Ihre Aufgabe besteht darin, die Wirkung dieser DVV auf die Datenstruktur und auf nachgelagerte Data-Mining-Verfahren messbar zu machen. Dabei sind sie integraler Bestandteil des methodischen Designs, das eine systematische und objektive Vergleichbarkeit verschiedener Verfahren und Verfahrenskombinationen ermöglicht. Da sich die Auswirkungen von DVV häufig nicht direkt, sondern nur im Zusammenspiel mit nachgelagerten Data-Mining-Verfahren zeigen, erfüllen Evaluationsmetriken eine vermittelnde Funktion. Sie verbinden die abstrakte Zielsetzung der Vorverarbeitung, etwa die Verbesserung der Analysequalität oder die Reduktion unnötiger Komplexität, mit konkreten und überprüfbaren Ergebnissen. Dies erfordert Bewertungsgrößen, die sowohl sensibel für Veränderungen in den Eingangsdaten als auch robust gegenüber externen Einflussfaktoren sind. Nur so lässt sich sicherstellen, dass beobachtete Unterschiede tatsächlich auf die gewählten DVV zurückzuführen sind und nicht durch modellinterne Eigenschaften oder zufällige Fluktuationen überlagert werden.

Das Ziel der Verwendung von Evaluationsmetriken besteht darin, differenzierte Aussagen über den Nutzen einzelner Verfahren, deren Kombinationen sowie über die Reihenfolge ihrer Anwendung zu ermöglichen. Vor allem im Hinblick auf Reproduzierbarkeit und Übertragbarkeit der Ergebnisse kommt der Auswahl geeigneter Metriken eine hohe Bedeutung zu. Sie steuern nicht nur die Interpretation der Resultate, sondern wirken sich auf die Struktur und Durchführung des gesamten Experimentierplans aus. Darüber hinaus dienen Evaluationsmetriken in dieser Arbeit nicht nur der nachträglichen Bewertung der Vorverarbeitungsergebnisse, sondern werden bereits in der Planungsphase berücksichtigt. Auf diese Weise tragen sie dazu bei, den Experimentierplan methodisch abzusichern und eine fundierte Grundlage für die spätere Analyse zu schaffen. Ihre Auswahl trägt dazu bei, vergleichbare Rahmenbedingungen zu definieren, unter denen verschiedene Verfahren, Verfahrenskombinationen und Reihenfolgen systematisch analysiert werden können. Durch diese vorausschauende Integration leisten sie einen wichtigen Beitrag zur Konzeption und Strukturierung des Untersuchungsdesigns. Gleichzeitig bilden sie die Grundlage für methodische Entscheidungen, etwa zur Auswahl geeigneter Verfahrenskombinationen oder zur Bewertung von Effekten bei variierenden Datenkonfigurationen. Denn nur durch geeignete Metriken lassen sich Unterschiede in der Analysequalität systematisch quantifizieren und DVV objektiv vergleichen.

Die Auswahl geeigneter Metriken erfolgt dabei nicht willkürlich, sondern basiert auf klar definierten Anforderungen, die ihre methodische Eignung sicherstellen. Die systematische Auswahl geeigneter Bewertungsgrößen setzt daher eine methodische Fundierung voraus. Im folgenden Abschnitt werden die dafür relevanten Anforderungen diskutiert, die Evaluationsmetriken im Rahmen dieser Arbeit erfüllen müssen.

5.1.2 Methodische Anforderungen an Evaluationsmetriken

Die Auswahl geeigneter Evaluationsmetriken (vgl. Abschnitt 2.2.2) basiert auf methodischen Prinzipien, die sicherstellen sollen, dass die gewählten Bewertungsgrößen den spezifischen Anforderungen an die Bewertung von DVV gerecht werden. Eine Metrik ist in diesem Zusammenhang nur dann sinnvoll einsetzbar, wenn sie eine objektive, konsistente und nachvollziehbare Beurteilung ermöglicht und sich nicht von äußeren oder nachgelagerten Faktoren wie Modellwahl, Datenpartitionierung oder Implementierungsdetails verzerren lässt.

Damit Evaluationsmetriken zur Bewertung von DVV geeignet sind, müssen sie die relevanten Eigenschaften der Verfahren erfassen. Bspw. sollte eine Metrik zur Bewertung der Modellgüte Rückschlüsse auf die durch die Datenvorverarbeitung veränderte Generalisierungsfähigkeit eines Modells ermöglichen. Ebenso müssen Effizienzmetriken wie Rechenzeit oder Speicherverbrauch die tatsächliche technische Belastung eines Verfahrens widerspiegeln und dürfen nicht durch irrelevante Faktoren verzerrt werden. Wenn die Aussagekraft einer Metrik unklar bleibt oder stark vom Anwendungskontext abhängt, ist eine methodisch abgesicherte Untersuchung nicht möglich. Die Fähigkeit einer Metrik, im Kontext der Datenvorverarbeitung genau diese inhaltlich relevanten Aspekte zu erfassen, wird im Folgenden als inhaltliche Validität bezeichnet.

Für eine methodisch fundierte Untersuchung ist es außerdem entscheidend, dass Evaluationsmetriken Ergebnisse liefern, die über verschiedene Verfahren und Datenzustände hinweg konsistent interpretiert werden können. Das bedeutet, dass sie unter gleichen Bedingungen zu stabilen und vergleichbaren Ergebnissen führen müssen und nicht durch zufällige Schwankungen verzerrt werden dürfen. Vergleichbarkeit setzt außerdem voraus, dass die experimentellen Rahmenbedingungen kontrolliert und konstant sind, etwa hinsichtlich Modellwahl und Randomisierung. Nur so lassen sich DVV methodisch trennscharf beurteilen. Die Fähigkeit von Metriken, Vergleiche zwischen Verfahren zu ermöglichen, wird im Folgenden als Vergleichbarkeit bezeichnet.

Weiterhin sollen Evaluationsmetriken in dieser Arbeit die Wirkung der Datenvorverarbeitung erfassen und nicht durch die Wahl eines bestimmten Klassifikators oder durch softwareseitige Besonderheiten verfälscht werden. Andernfalls ließe sich nicht eindeutig feststellen, ob beobachtete Unterschiede auf die Datenvorverarbeitung oder auf modellspezifische Effekte zurückzuführen sind. Aus diesem Grund wird als methodische Anforderung an Evaluationsmetriken die Unabhängigkeit von Modellarchitektur und Implementierung herangezogen.

Evaluationsmetriken können nur dann sinnvoll eingesetzt werden, wenn ihre Berechnung mit vertretbarem Rechen- und Implementierungsaufwand möglich ist Dazu gehört, dass eine Metrik effizient berechenbar und in verbreiteten Open-Source-Bibliotheken verfügbar ist. Der Einsatz etablierter Python-Bibliotheken (Python Software Foundation, 2023) trägt außerdem dazu bei, eine standardisierte und reproduzierbare Umsetzung sicherzustellen. Metriken, deren Berechnung mit erheblichem technischem oder rechnerischem Aufwand verbunden ist, sind für eine methodisch fundierte Untersuchung nicht geeignet. Diese Anforderung wird im Folgenden als technische Umsetzbarkeit bezeichnet.

Damit Evaluationsmetriken in der Praxis einen Mehrwert bieten, müssen ihre Ergebnisse für Anwender verständlich und eindeutig interpretierbar sein. Nur so lassen sich Schlussfolgerungen über die Wirkung einzelner DVV oder ganzer Pipelines ziehen. Voraussetzung dafür ist, dass Metriken eindeutig definiert sind, beispielsweise durch klare mathematische Formeln, und somit eine transparente sowie konsistente Anwendung erlauben. Auf diese Weise bleibt insbesondere bei der Analyse unterschiedlicher Kombinationen und Reihenfolgen von Verfahren sichergestellt, dass beobachtete Effekte klar zugeordnet werden können. Dieses Kriterium

wird als Praktikabilität und Interpretierbarkeit bezeichnet. Die nachfolgende Tabelle 5.1 fasst die methodischen Anforderungen an Evaluationsmetriken sowie deren jeweilige Bedeutung kompakt zusammen.

Tabelle 5.1: Methodische Anforderungen an Evaluationsmetriken

Kriterium	**Bedeutung**
Inhaltliche Validität	Metrik muss tatsächlich das bewerten, was durch die Datenvorverarbeitung beeinflusst wird
Vergleichbarkeit	Metrik muss zwischen verschiedenen Verfahren und unter konstanten Bedingungen vergleichbare Aussagen treffen
Modellunabhängigkeit	Bewertung darf nicht vom eingesetzten Modell oder der Softwareumgebung abhängen
Technische Umsetzbarkeit	Berechnung muss effizient, nachvollziehbar und mit verfügbaren Werkzeugen realisierbar sein
Praktikabilität und Interpretierbarkeit	Metrik muss verständlich sein und sinnvolle Rückschlüsse auf die Wirkung von Datenvorverarbeitung erlauben

Tabelle 5.1 fasst die zuvor erläuterten Kriterien zusammen und verdeutlicht, welche Anforderungen Evaluationsmetriken im Kontext dieser Arbeit erfüllen müssen. Sie dient damit als strukturierte Orientierung für die methodische Auswahl geeigneter Metriken.

Diese Anforderungen bilden die Grundlage für eine belastbare und nachvollziehbare Bewertung von DVV. Im nächsten Schritt werden die übergeordneten Bewertungsdimensionen eingeführt, anhand derer sich die Wirkung einzelner DVV systematisch analysieren lässt.

5.1.3 Ermittlung geeigneter Bewertungsdimensionen

Bevor konkrete Evaluationsmetriken ausgewählt werden können, müssen zunächst geeignete Bewertungsdimensionen für die Beurteilung von DVV festgelegt werden. Bewertungsdimensionen bezeichnen in diesem Kontext übergeordnete qualitative Aspekte, die für datenanalytische Verfahren von Bedeutung sind und durch spezifische Metriken messbar gemacht werden können. Die Auswahl geeigneter Bewertungsdimensionen orientiert sich an den methodischen Anforderungen, die im vorhergehenden Abschnitt definiert wurden. Die Festlegung dieser Dimensionen stellt einen wichtigen Zwischenschritt dar. Um passende Evaluationsmetriken auszuwählen, muss zunächst festgelegt werden, welche Aspekte der Performance von DVV untersucht werden sollen. Zudem ist zu klären, inwiefern sich deren Einfluss auf die Leistung eines nachgelagerten Data-Mining-Verfahrens erfassen lässt. Bewertungsdimensionen fungieren somit als inhaltliche Brücke zwischen den in Abschnitt 5.1.2 formulierten methodischen Anforderungen und der konkreten Auswahl geeigneter Metriken. Im Folgenden werden geeignete Bewertungsdimensionen identifiziert, die sich sowohl theoretisch begründen als auch praktisch umsetzen lassen. Grundlage dafür ist die Zielsetzung, die Wirkung von DVV nicht einseitig, sondern entlang unterschiedlicher Kriterien zu analysieren.

Die Modellgüte gilt, wie in Abschnitt 2.2.2 in nahezu allen datengetriebenen Verfahren als zentrale Bewertungsdimension, da sie die Fähigkeit eines Analysemodells beschreibt, auf Basis gelabelter Daten korrekte Vorhersagen zu treffen und diese Leistung auf unbekannte Daten zu übertragen. Im Kontext der Datenvorverarbeitung ist dies besonders relevant, da jegliche Veränderungen der Datenstruktur, wie etwa durch Behandlung fehlender Werte, Merkmals- oder Instanzenreduktion, direkte Auswirkungen auf die Eingabedaten des Modells haben. Dadurch verändern sich potenziell Muster in den Eingabedaten, die das Modell beim Training

erfasst, was wiederum die Modelleigenschaften wie etwa Genauigkeit oder Generalisierungsfähigkeit beeinflussen kann. Die Herausforderung besteht dabei in der Tatsache, dass Modellgüte nicht durch die Datenvorverarbeitung isoliert entsteht, sondern im Zusammenspiel mit dem gewählten Analysemodell und der konkreten Anwendungssituation. Deshalb wird in dieser Arbeit ein einheitlicher, verbreiteter Klassifikator als Werkzeug eingesetzt, um Veränderungen in der Modellleistung konsistent zu erfassen. Die Analyse folgt dabei einem Blackbox-Prinzip, bei dem nicht die einzelnen DVV im Detail betrachtet werden, sondern deren Auswirkungen auf die Ergebnisse des Klassifikators. Im Mittelpunkt steht somit das Gesamtsystem, dessen Leistungsfähigkeit sich darin widerspiegelt, wie sich die Datenvorverarbeitung auf die Vorhersagegüte des Modells auswirkt. Zur Messung der Modellgüte stehen etablierte Kennzahlen wie Accuracy, Precision, Recall und FM zur Verfügung, die bereits in Abschnitt 2.2.2 eingeführt wurden. Diese Metriken ermöglichen eine differenzierte Betrachtung verschiedener Fehlertypen, z. B. falsch-positive und falsch-negative Klassifikationen, und erlauben dadurch eine differenzierte Bewertung der Wirkung von DVV. Die Modellgüte erfüllt damit alle methodischen Anforderungen, die zuvor definiert wurden. Sie ist demnach inhaltlich valide, über verschiedene Verfahren hinweg vergleichbar, unabhängig von Implementierungsdetails und mit geringem technischen Aufwand messbar. Ihre Einbeziehung als Bewertungsdimension erlaubt es, die Effektivität der Datenvorverarbeitung direkt am Zielkriterium datengetriebener Verfahren zu messen, der Leistungsfähigkeit eines Analysemodells.

Die Effizienz eines DVV bezieht sich, wie in Abschnitt 2.2.2 beschrieben, auf den Ressourcenaufwand, der mit dessen Anwendung verbunden ist. Dazu zählen typischerweise Rechenzeit und Speicherverbrauch. Diese Aspekte können in datenintensiven Szenarien eine wesentliche Rolle spielen, insbesondere wenn DVV skaliert oder regelmäßig automatisiert ausgeführt werden müssen. Die Relevanz der Effizienz hängt maßgeblich davon ab, welche Anforderungen ein konkreter Anwendungskontext an Ressourcenverbrauch, Laufzeit und Umsetzbarkeit stellt. Je nach verfügbarer Infrastruktur und erwarteter Verarbeitungszeit kann die Bedeutung dieser Dimension erheblich variieren. In dieser Arbeit, die sich an datengetriebenen Analyseprozessen in einem industriellen Kontext orientiert, wird davon ausgegangen, dass die technische und zeitliche Effizienz ein wichtiger Faktor ist. Es stellt sich die Frage, ob Effizienz lediglich als praktischer Begleitfaktor zu verstehen ist oder als eigenständige Bewertungsdimension betrachtet werden sollte. Für eine eigenständige Betrachtung spricht hingegen, dass Effizienzgewinne auch dann von Bedeutung sein können, wenn sie nicht mit einer unmittelbaren Verbesserung der Modellgüte einhergehen. Gerade im Vergleich verschiedener Vorverarbeitungspipelines kann die explizite Berücksichtigung der Effizienz dazu beitragen, die Gesamtleistung differenzierter und anwendungsnäher zu bewerten. Darüber hinaus erfüllt die Effizienzdimension grundlegende methodische Anforderungen. Sie ist demnach unabhängig von der Modellwahl, empirisch gut messbar und erlaubt eine transparente Einschätzung des Verhältnisses von Aufwand und Ertrag. Aufgrund ihrer praktischen Relevanz und methodischen Robustheit wird Effizienz in dieser Arbeit als eigenständige Bewertungsdimension berücksichtigt. Die relevanten Metriken zur Bewertung der Effizienz in dieser Arbeit werden in Abschnitt 5.1.4 ausgewählt.

Die Robustheit als Bewertungsdimension bezieht sich auf die Fähigkeit eines DVV, auch bei leichten Störungen oder Variationen in den Eingangsdaten stabile Analyseergebnisse zu ermöglichen, wie in Abschnitt 2.2.2 dargelegt wurde. Gerade im Kontext dieser Arbeit, in der Verfahren zur Behandlung fehlender Werte sowie zur Reduktion von Daten untersucht werden, erscheint die Robustheit besonders relevant. Denn beide Arten von Datenvorverarbeitung greifen in die Datenstruktur ein, sei es durch das Ergänzen nicht vorhandener Informationen oder durch das gezielte Entfernen vermeintlich irrelevanter Merkmale oder Instanzen (vgl. Kapitel 3, 4). In beiden Fällen besteht die Gefahr, dass sich kleine Änderungen in den Eingabedaten deutlich auf das Analyseergebnis auswirken. In der Praxis sind idealtypische Datenbedingungen selten gegeben, wie bereits in Abschnitt 2.3 deutlich wurde. Abweichungen durch Ausreißer, fehlerhafte Sensorwerte oder zufällige Lücken in der Datenerfassung sind demnach typische Herausforderungen, die in industriellen Anwendungen auftreten (vgl. Abschnitt 2.3). Verfahren, die auf bestimmte Verteilungen oder homogene Datenverhältnisse angewiesen sind,

können unter solchen Bedingungen instabil reagieren. Vor diesem Hintergrund liefert die Bewertung der Robustheit wichtige Hinweise auf die Übertragbarkeit und Belastbarkeit des jeweiligen Verfahrens. Dabei ist allerdings zu bedenken, dass Robustheit nicht vollständig unabhängig von anderen Bewertungsdimensionen ist, da sich Instabilität häufig in einer erhöhten Varianz der Modellleistung widerspiegelt. Im Ergebnis wird Robustheit in dieser Arbeit als Bewertungsdimension berücksichtigt. Sie ist insbesondere im Zusammenspiel mit den untersuchten Verfahren zur Imputation und Datenreduktion geeignet, auf methodische Schwächen oder Abhängigkeiten hinzuweisen, die sich in der reinen Modellgüte nicht hinreichend zeigen. Ihre Relevanz ergibt sich aus ihrer Fähigkeit, die Stabilität datenverändernder Verfahren unter realitätsnahen Bedingungen differenziert zu beurteilen.

Andere Bewertungsdimensionen, wie die in Abschnitt 2.2.2 angerissene Fairness werden im Rahmen dieser Arbeit nicht berücksichtigt. Zwar ist sie in bestimmten Anwendungsfeldern durchaus relevant, insbesondere wenn es personenbezogene Daten und diskriminierungsfreie Entscheidungen geht. Allerdings erfüllt sie nicht die methodischen Anforderungen, die an die Auswahl der Bewertungsdimensionen in dieser Arbeit gestellt werden. Fairness stellt eine eigenständige ethische Bewertungsdimension dar, deren methodische Operationalisierung spezifische Kontextkenntnisse und gesellschaftliche Bewertungsmaßstäbe erfordert. Da diese Arbeit keine Anwendung mit sensiblen personenbezogenen Daten behandelt, ist eine Integration entsprechender Metriken zur Messung von Fairness nicht vorgesehen. Insgesamt lässt sich festhalten, dass diese Bewertungsdimension zwar grundsätzlich zu berücksichtigen ist, im Rahmen der vorliegenden Fragestellung und unter den gegebenen methodischen Voraussetzungen jedoch nicht bewertet werden kann.

Die Analyse der zur Auswahl stehenden Bewertungsdimensionen verdeutlicht, dass insbesondere Modellgüte, Effizienz und Robustheit sowohl den methodischen Anforderungen dieser Arbeit entsprechen als auch inhaltlich eine differenzierte Bewertung von Verfahren zur Behandlung fehlender Werte und zur Datenreduktion ermöglichen. Diese drei Dimensionen bilden ein ausgewogenes Spektrum, das sowohl die Leistungsfähigkeit, die technische Umsetzbarkeit als auch die Stabilität der Verfahren unter realitätsnahen Bedingungen adressiert. Aufgrund ihrer theoretischen Fundierung, praktischen Relevanz und experimentellen Zugänglichkeit werden sie im Rahmen des folgenden Experimentierplans als zentrale Bewertungsdimensionen herangezogen, auf deren Basis nachfolgend die Auswahl konkreter Evaluationsmetriken erfolgt.

5.1.4 Auswahl geeigneter Evaluationsmetriken

Aufbauend auf den zuvor entwickelten Bewertungsdimensionen erfolgt in diesem Abschnitt die methodisch fundierte Auswahl konkreter Evaluationsmetriken. Ziel ist es, solche Metriken zu identifizieren, die die drei ausgewählten Bewertungsdimensionen jeweils differenziert, valide und experimentell zuverlässig erfassen. Die Auswahl erfolgt dabei unter Bezugnahme auf die in Abschnitt 5.1.2 formulierten methodischen Anforderungen an Evaluationsmetriken.

Modellgüte

Als geeignete Metriken werden Accuracy, Precision, Recall und FM herangezogen (vgl. Abschnitt 2.2.2). Diese Auswahl erfüllt die Anforderung der inhaltlichen Validität**,** da jede dieser Metriken gezielt Aspekte der Modellleistung quantifiziert, die durch Datenvorverarbeitung beeinflusst werden können. Die Accuracy gibt Aufschluss über den Gesamtanteil korrekt klassifizierter Instanzen, ist jedoch anfällig gegenüber unausgeglichenen Klassenverteilungen. Zur differenzierteren Bewertung werden Precision und Recall herangezogen, die zwischen falsch-positiven und falsch-negativen Klassifikationen unterscheiden. Die FM als harmonisches Mittel dieser beiden Metriken ermöglicht eine ausgewogene Bewertung, insbesondere bei ungleichen Klassenverteilungen.

Die Vergleichbarkeit der gewählten Metriken ist durch ihre normierte Skalierung und verbreitete Anwendung gegeben. Sie lassen sich unabhängig von der konkreten Datensituation auf unterschiedliche DVV anwenden und ermöglichen konsistente Aussagen über deren Wirkung.

Die Modellunabhängigkeit wird sichergestellt, indem ein konstanter Klassifikator mit festen Parametern verwendet wird. Dadurch ist gewährleistet, dass Unterschiede in den Metriken auf die Datenvorverarbeitung und nicht auf Schwankungen im Modellverhalten zurückzuführen sind. Diese Kontrolle über die Modellkonfiguration ermöglicht eine methodisch saubere Bewertung der Vorverarbeitungseffekte. Hinsichtlich der technischen Umsetzbarkeit erfüllen alle gewählten Metriken die geforderten Kriterien. Sie sind vollständig in gängigen Bibliotheken wie `scikit-learn` (scikit-learn developers, 2023) implementiert, effizient berechenbar und gut dokumentiert. Konkret können Accuracy, Precision, Recall und FM mithilfe der Funktionen `accuracy_score`, `precision_score`, `recall_score` und `f1_score` aus dem Modul `sklearn.metrics` direkt berechnet werden. Dies ermöglicht ihre nahtlose Integration in die Versuche sowie eine automatisierte und reproduzierbare Erhebung der Bewertungsergebnisse. Die gewählten Metriken sind in der Praxis etabliert und liefern Kennzahlen, die sowohl inhaltlich differenziert als auch kommunikativ anschlussfähig sind (vgl. Abschnitt 2.2.2). Sie sind für verschiedene Anwendergruppen nachvollziehbar und ermöglichen eine transparente Bewertung der Auswirkungen einzelner DVV auf die Modellgüte. Damit werden auch die Anforderungen an Praktikabilität und Interpretierbarkeit erfüllt.

Zur Erhebung der genannten Metriken wird in allen Versuchen eine Kreuzvalidierung eingesetzt. Dieses Verfahren erhöht die Verlässlichkeit der Ergebnisse, da nicht nur eine einzelne zufällige Aufteilung in Trainings- und Testdaten betrachtet wird, sondern mehrere Wiederholungen erfolgen (vgl. Abschnitt 2.2.2). Dadurch wird das Risiko reduziert, dass die Bewertung der Modellgüte stark von einer bestimmten Datenteilung abhängt, und es entsteht eine robustere Grundlage für den Vergleich der DVV.

Insgesamt bilden Accuracy, Precision, Recall und FM ein wissenschaftlich anerkanntes, methodisch begründetes und technisch umsetzbares Instrumentarium zur Bewertung der Modellgüte im Kontext datenbasierter Vorverarbeitung. Ihre Kombination erlaubt es, die differenzierte Wirkung von Verfahren zur Behandlung fehlender Werte und zur Datenreduktion umfassend und belastbar zu analysieren.

Effizienz

Die Effizienz eines DVV beschreibt den Ressourcenaufwand, der mit seiner Anwendung verbunden ist (vgl. Abschnitt 2.2.2). Sie ist insbesondere dann relevant, wenn Verfahren unter zeitkritischen Bedingungen eingesetzt, in automatisierte Pipelines integriert oder regelmäßig neu ausgeführt werden sollen. Für die vorliegende Arbeit, die auf eine methodisch fundierte Vergleichbarkeit von DVV zielt, wird die Effizienz als eigene Bewertungsdimension berücksichtigt. Die Auswahl geeigneter Metriken zur Bewertung der Effizienz erfolgt unter Berücksichtigung der in Abschnitt 5.1.2 definierten methodischen Anforderungen. Als zentrale Metrik wird die Rechenzeit herangezogen, da sie eine direkt beobachtbare und technisch zuverlässig messbare Größe darstellt.

Die Rechenzeit ist inhaltlich valide, da sie den unmittelbaren zeitlichen Aufwand eines Verfahrens erfasst und damit Rückschlüsse auf dessen praktische Einsetzbarkeit ermöglicht. Die Messung wird jeweils für den gesamten Ablauf einer Versuchskonfiguration durchgeführt, beginnend vor der Datenvorverarbeitung und endend nach Abschluss der Kreuzvalidierung. Auf diese Weise umfasst die Rechenzeit sowohl die DVV als auch das Training und die Evaluation des Modells und ermöglicht eine ganzheitliche Einschätzung des zeitlichen Aufwands. Die Rechenzeit ist außerdem vergleichbar, da sie unter kontrollierten Bedingungen und bei identischer Infrastruktur für alle Verfahren erhoben wird. Dieses Vorgehen erlaubt konsistente Aussagen über relative Unterschiede im Ressourcenverbrauch. Die Vergleichbarkeit ist gewährleistet, da für alle Konfigurationen derselbe Klassifikator mit identischen Parametern eingesetzt wird. Unterschiede in der Rechenzeit lassen sich somit eindeutig auf die Datenvorverarbeitung zurückführen. Bezüglich der technischen Umsetzbarkeit ist die Rechenzeit unproblematisch zu erfassen. Sie erfordert keine zusätzlichen Systemvoraussetzungen, ist mithilfe des in Python integrierten Moduls `time` messbar und lässt sich leicht in bestehende Workflows integrieren. Auch die Praktikabilität und Interpretierbarkeit ist hoch, denn die Kennzahl ist intuitiv

verständlich, unmittelbar nachvollziehbar und liefert eine konkrete Grundlage für die Bewertung der Effizienz eines Verfahrens.

Auf die zusätzliche Messung des Speicherverbrauchs wird in dieser Arbeit bewusst verzichtet. Zwar kann der Speicherbedarf bei datenintensiven Anwendungen von Bedeutung sein, jedoch ist er im vorliegenden Anwendungsrahmen mit mittelgroßen Datensätzen und definierter Umgebung kein entscheidender Faktor. Zudem ist der Speicherverbrauch stark abhängig von systemexternen Prozessen und weist eine geringere Messstabilität auf. Da sich aus der Rechenzeit bereits eine belastbare Aussage zur operativen Effizienz ableiten lässt, wird die Einbeziehung des Speicherverbrauchs aus methodischen und praktischen Gründen nicht weiterverfolgt. Insgesamt stellt die Rechenzeit eine robuste, vergleichbare und in der Praxis relevante Metrik dar, um die Effizienz von DVV im Rahmen des Experimentierplans differenziert zu bewerten.

Robustheit

Robustheit wird in dieser Arbeit als wichtige Bewertungsdimension berücksichtigt, da die untersuchten DVV in realen, oft fehleranfälligen Datenumgebungen zuverlässig funktionieren müssen. Besonders bei Imputation und Datenreduktion ist es entscheidend, dass Verfahren auch unter instabilen Datenbedingungen verlässlich funktionieren. Die Erfassung der Robustheit orientiert sich an den methodischen Anforderungen aus Abschnitt 5.1.2. In dieser Arbeit wird Robustheit über die Standardabweichung der Modellgütemaße definiert, die sich aus den Wiederholungen der Kreuzvalidierung ergeben. Konkret dienen die Standardabweichungen der in den Folds ermittelten Kennzahlen, etwa der Accuracy, als Indikator für die Stabilität der jeweiligen Verfahren (vgl. Abschnitt 2.2.2). Auf diese Weise wird sichtbar, ob die Ergebnisse stark von der zufälligen Aufteilung der Daten abhängen oder ob ein Verfahren auch bei variierenden Trainings- und Testmengen stabile Resultate liefert.

Die gewählte Herangehensweise erfüllt die Anforderung der inhaltlichen Validität, da sie die Empfindlichkeit der Verfahren gegenüber unvermeidbaren Zufallseffekten in realen Datensituationen abbildet. Die Vergleichbarkeit ist gegeben, da alle Verfahren unter identischen Rahmenbedingungen bewertet werden und dieselbe Kreuzvalidierung angewendet wird. Die Messung ist außerdem modellunabhängig, da stets derselbe Klassifikator verwendet wird und nur die Vorverarbeitungsbedingungen modifiziert werden. Auch die technische Umsetzbarkeit ist gewährleistet, da die benötigten Wiederholungen und Variationen sich mit überschaubarem Rechenaufwand in bestehenden Python-Workflows implementieren lassen. Standardisierte Zufallssteuerung über der Parameter `random_state` der `scikit-learn`-Bibliothek sorgt dabei für die Reproduzierbarkeit der Ergebnisse. Bezüglich der Praktikabilität und Interpretierbarkeit bietet die Varianz der Metriken zur Bewertung der Modellgüte eine intuitive und breit anerkannte Grundlage zur Bewertung von Stabilität. Sie macht sichtbar, ob ein Verfahren zuverlässig arbeitet oder empfindlich auf geringfügige Veränderungen reagiert. Damit liefert die Robustheitsbewertung zusätzliche Erkenntnisse, die über die reine Leistung unter Idealbedingungen hinausgehen. Die gewählte Umsetzung stellt sicher, dass die Robustheit datenverändernder Maßnahmen empirisch belastbar, methodisch nachvollziehbar und technisch praktikabel beurteilt werden kann. In Kombination mit Modellgüte und Effizienz ergibt sich ein ganzheitliches Bewertungsprofil für die untersuchten DVV. Die folgende Tabelle 5.2 zeigt die ausgewählten Evaluationsmetriken sowie ihre jeweilige Bewertungsdimension und Funktion.

Die in Tabelle 5.2 zusammengefassten Metriken bilden die Grundlage für die experimentelle Bewertung der DVV im Rahmen des Experimentierplans. Ihre Zuordnung zu den jeweiligen Bewertungsdimensionen ermöglicht eine systematische und nachvollziehbare Analyse der Effekte unterschiedlicher Strategien. Durch die Kombination etablierter Metriken mit klaren Zuordnungskriterien entsteht ein konsistenter Bewertungsrahmen, der sowohl methodischen als auch praktischen Anforderungen gerecht wird.

Da eine vollständige Berücksichtigung aller erhobenen Metriken zu einer hohen Komplexität und eingeschränkten Übersichtlichkeit führen würde, ist es sinnvoll, die Auswertung zunächst auf primäre Metriken zu fokussieren. Diese sollten die wesentlichen Aspekte der Modellgüte und Effizienz abbilden und zugleich eine kompakte, vergleichbare Darstellung der Ergebnisse

Tabelle 5.2: Auswahl der Evaluationsmetriken

Bewertungsdimension	Evaluationsmetrik	Funktion
Modellgüte	Accuracy	Messung der Gesamtleistung
Modellgüte	Precision	Erkennung falsch-positiver Klassifikationen
Modellgüte	Recall	Erkennung falsch-negativer Klassifikationen
Modellgüte	FM	Balance der Fehlerarten
Effizienz	Rechenzeit	Messung des Zeitaufwands
Robustheit	Standardabweichung der Modellgüte	Messung der Stabilität

ermöglichen. Accuracy beschreibt die Gesamtgüte des Modells, während die FM Precision und Recall zu einer ausgewogenen Maßzahl verdichtet und damit eine kompakte Aussage über beide Metriken liefert. Die Rechenzeit ergänzt diese Betrachtung um die Dimension der Effizienz. Vor diesem Hintergrund dienen Accuracy und die FM als primäre Maße zur Bewertung der Modellgüte sowie die Rechenzeit zur Erfassung der Effizienz. Precision, Recall und die Standardabweichungen der Gütemaße werden zwar ebenfalls erhoben, jedoch als sekundäre Metriken betrachtet. Die Aussagen von Precision und Recall sind bereits in der FM enthalten, während die Standardabweichungen für jede Gütemetrik einzeln vorliegen und daher nicht isoliert, sondern im Gesamtbild als Maß für Stabilität und Robustheit der Verfahren eingeordnet werden. Alle ausgewählten Metriken werden vollständig dokumentiert und für detailliertere Analysen herangezogen. Somit bleibt gewährleistet, dass trotz Fokussierung auf primäre Metriken auch die sekundären Metriken in die Analyse einbezogen werden können. Im nächsten Abschnitt erfolgt die Auswahl konkreter DVV, die im Rahmen des Experimentierplans untersucht werden.

5.2 Kriterien zur Auswahl geeigneter Datenvorverarbeitungsverfahren

Die Auswahl geeigneter DVV bildet einen methodisch bedeutsamen Zwischenschritt dieser Arbeit. Sie ist notwendig, um in einem späteren Experimentierplan eine fundierte und systematisch begründete Untersuchung der Wirkung der angewandten Verfahren durchführen zu können. Ziel dieses Abschnitts ist es, Auswahlkriterien zu definieren, die eine transparente, kontextgerechte und objektiv nachvollziehbare Eingrenzung der im Experiment zu untersuchenden Verfahren ermöglichen. Die konkrete Auswahl erfolgt im Implementierungskapitel und orientiert sich an den hier festgelegten Kriterien. Die Kriterien leiten sich aus den methodischen Zielsetzungen dieser Arbeit sowie aus inhaltlichen Bewertungen der in den Abschnitten 3.3, 0 und 4.3 dargestellten Verfahren ab. Dabei werden sowohl Verfahren zur Behandlung fehlender Werte als auch solche zur Datenreduktion auf Merkmals- und Instanzebene berücksichtigt.

Die Auswahl der Verfahren im Rahmen des Experimentierplans richtet sich maßgeblich nach ihrer Eignung für realistische Datenprobleme. Besonders geeignet sind Verfahren, die mit typischen Merkmalen industrieller Daten wie fehlenden Sensorwerten, Redundanzen, hoher Dimensionalität oder inkonsistenten Eingaben umgehen können. Verfahren hingegen, die ausschließlich unter idealisierten Voraussetzungen zuverlässig funktionieren oder eine vollständige Datenmatrix voraussetzen, sind für den Anwendungsrahmen dieser Arbeit nur eingeschränkt geeignet. Ebenfalls entscheidend ist die Robustheit gegenüber Variationen in den Eingangsdaten. Verfahren, deren Resultate stark schwanken, wenn sich z. B. der Anteil fehlender Werte leicht ändert, oder die instabil auf unterschiedliche Partitionierungen reagieren,

können im Rahmen einer empirisch vergleichenden Studie nur eingeschränkt interpretiert werden. In Abschnitt 3.3.5 wurde etwa diskutiert, dass bestimmte multivariate Verfahren sensitiv auf Datenverteilungen reagieren. Dies stellt eine Eigenschaft dar, die bei der Auswahl der Verfahren berücksichtigt werden muss. Da im Experimentierplan nicht nur die Wirkung einzelner Schritte, sondern auch Kombinations- und Reihenfolgeneffekte untersucht werden, müssen die eingesetzten Verfahren unabhängig voneinander anwendbar sein. Verfahren, die interne Modellannahmen voraussetzen oder als vollständige Pipelines agieren, erschweren eine gezielte Variation und methodische Isolierbarkeit. Dies betrifft insbesondere komplexe Wrapper- (vgl. Abschnitt 4.2.4) oder Embedded-Verfahren (vgl. Abschnitt 4.2.5), bei denen Auswahl und Modelltraining eng verzahnt sind. Vor diesem Hintergrund ist die Modularität und Kombinierbarkeit der Verfahren ein wesentliches Kriterium für ihre Auswahl. Verfahren mit hohem Laufzeit- oder Speicheraufwand, wie etwa modellbasierte Imputation oder Wrapper-basierte Merkmalsselektion, sind nur dann sinnvoll einsetzbar, wenn ihr Mehrwert gegenüber effizienteren Ansätzen hinreichend groß ist. In Abschnitt 3.3.6 wurde gezeigt, dass bestimmte komplexe Imputationsverfahren zwar eine hohe Genauigkeit versprechen, unter ressourcenkritischen Bedingungen jedoch schwer skalierbar sind. Im Sinne der Praktikabilität und Umsetzbarkeit müssen solche Verfahren daher kritisch abgewogen werden. Vor diesem Hintergrund bildet die Rechenökonomie und Skalierbarkeit ein zentrales Kriterium für die Auswahl geeigneter Verfahren. Gerade in einem experimentellen Vergleich, in dem viele Schritte methodisch nachvollzogen und dokumentiert werden müssen, ist es außerdem erforderlich, dass die Wirkung eines Verfahrens verständlich und prinzipiell nachvollziehbar bleibt. Verfahren, die auf undokumentierten oder schwer zugänglichen Heuristiken beruhen, sind daher aus methodischen Gründen weniger geeignet. Vor diesem Hintergrund stellt die Interpretierbarkeit der Verfahren und ihrer Wirkung ein zentrales Auswahlkriterium dar. Damit die Verfahren zuverlässig angewendet und in bestehende Evaluationsprozesse integriert werden können, müssen sie mit verbreiteten Python-Bibliotheken umsetzbar sein und eine stabile Programmierschnittstelle aufweisen. Ansätze, die auf experimentellen Implementierungen ohne dokumentierte Stabilität beruhen, sind dagegen für eine systematische Auswertung ungeeignet. Vor diesem Hintergrund ist die technische Verfügbarkeit und Stabilität der Implementierung ein wesentliches Auswahlkriterium.

Die genannten Kriterien bilden den Rahmen für die spätere Auswahl geeigneter Verfahren im Implementierungskapitel. Sie gewährleisten, dass die Auswahl methodisch fundiert, empirisch belastbar und theoretisch anschlussfähig erfolgt unter Rückgriff auf die im Grundlagenteil erarbeiteten Differenzierungen und in direkter Orientierung an den Zielstellungen des Experimentierplans. Die Tabelle 5.3 konkretisiert die in diesem Abschnitt hergeleiteten Auswahlkriterien und fasst die zentralen Anforderungen an geeignete DVV kompakt zusammen.

Tabelle 5.3: Kriterien zur Auswahl von DVV

Kriterium	**Bedeutung**
Eignung für reale Datenprobleme	Verfahren müssen unter industriellen Bedingungen einsetzbar sein
Robustheit gegenüber Datenvariationen	Stabile Ergebnisse erforderlich
Modularität und Kombinierbarkeit	Verfahren müssen isoliert und in Kombination anwendbar sein
Rechenökonomie und Skalierbarkeit	Ressourcenschonung für wiederholte Anwendung notwendig
Interpretierbarkeit	Funktionsweise muss nachvollziehbar sein
Verfügbarkeit in Python-Bibliotheken	Implementierung in stabilen, dokumentierten Bibliotheken erforderlich

Tabelle 5.3 dient als methodisch fundierter Bewertungsrahmen, anhand dessen potenzielle Verfahren im Hinblick auf ihre inhaltliche Eignung, technische Umsetzbarkeit und experimentelle Belastbarkeit systematisch geprüft werden können. Durch die strukturierte Gegenüberstellung der Kriterien entsteht eine nachvollziehbare Grundlage für die Auswahl der Verfahren, die in der Implementierung des Experimentierplans erfolgt. Aufbauend auf dieser Basis wird im nächsten Abschnitt der Experimentierplan konzeptioniert, der DVV gezielt kombiniert, in ihrer Anwendungsreihenfolge variiert und bewertet, um deren Wirkung unter kontrollierten Bedingungen methodisch fundiert zu analysieren.

5.3 Aufbau des Experimentierplans

Ziel dieses Abschnitts ist die Entwicklung eines experimentelles Designs, welches die differenzierte Bewertung von Verfahren zur Behandlung fehlender Werte und zur Datenreduktion ermöglicht. Im Fokus stehen dabei sowohl die Wirkung einzelner Verfahren als auch deren Kombinationen und Anwendungsreihenfolgen. Die Bewertung erfolgt anhand der in Abschnitt 5.1.3 bestimmten Bewertungsdimensionen Modellgüte, Effizienz und Robustheit. Der Experimentierplan bildet damit das methodische Bindeglied zwischen der theoretischen Fundierung und der praktischen Implementierung. Aufbauend auf dieser Zielsetzung werden im Folgenden die grundlegenden Bestandteile des Experimentierplans schrittweise entwickelt.

5.3.1 Konzeptionelle Zielsetzung und Bewertungsstrategie

Das Erkenntnisinteresse des Experiments liegt in der differenzierten Untersuchung der Wirkung verschiedener DVV auf die Leistungsfähigkeit eines nachgelagerten Data-Mining-Verfahrens. Betrachtet werden sowohl Verfahren zur Behandlung fehlender Werte als auch Maßnahmen zur Reduktion redundanter Merkmale und Instanzen. Da theoretische Überlegungen oder vereinzelte empirische Beobachtungen keine belastbaren Rückschlüsse auf das Zusammenwirken mehrerer Verarbeitungsschritte zulassen, ist ein systematisch aufgebauter Experimentierplan erforderlich. Erst durch gezielte Variation und kontrollierte Bewertung wird es möglich, isolierte Effekte, Wechselwirkungen und Reihenfolgeabhängigkeiten methodisch trennscharf zu erfassen. Ein weiterer Vorteil des experimentellen Aufbaus liegt in der Möglichkeit, nicht nur beschreibende, sondern auch kausale Aussagen über die Wirkung von Datenvorverarbeitung zu treffen. Die gezielte Durchführung einzelner Schritte in einem konstant gehaltenen Analyseumfeld erlaubt es, Ursache-Wirkungs-Zusammenhänge differenziert und nachvollziehbar darzustellen. Dabei wird berücksichtigt, dass die Performance datenverändernder Maßnahmen vom jeweiligen Datensatz abhängt. Der Experimentierplan ist so konzipiert, dass er kontextspezifisch erkennen lässt, welche Verfahren, Kombinationen und Reihenfolgen in einer konkreten Anwendung besonders gut geeignet sind.

Das Untersuchungsdesign folgt einem mehrstufigen Aufbau, der die verschiedenen Analyseperspektiven voneinander trennt. In einem ersten Schritt wird ein Analysemodell auf die Rohdaten ohne jegliche Datenvorverarbeitung angewendet. Diese Baseline dient als Ausgangspunkt zur Einordnung der späteren Ergebnisse und liefert eine Referenz für die Wirkung datenverändernder Maßnahmen. Anschließend werden die verschiedenen DVV einzeln untersucht. Dabei wird jedes Verfahren isoliert auf den unveränderten Datensatz angewendet und anschließend evaluiert, um seine spezifische Wirkung auf die Modellperformance festzustellen. In einem dritten Schritt erfolgt die kombinierte Anwendung von Verfahren, wobei auch deren Reihenfolge variiert wird. Diese Variation ermöglicht es, Wechselwirkungen zwischen Imputations- und Reduktionsmaßnahmen sichtbar zu machen und deren Auswirkungen auf die Analysequalität differenziert zu bewerten. Nach jeder Versuchspipeline wird ein konstantes Analysemodell auf die resultierenden Daten angewendet. Die Bewertung erfolgt jeweils anhand der in Abschnitt 5.1.4 definierten Metriken zur Modellgüte, Effizienz und Robustheit. Auf Zwischenmessungen nach Teilprozessen wird bewusst verzichtet, da alle betrachteten Verfahren zuvor bereits separat analysiert werden. Dieses Vorgehen gewährleistet eine saubere methodische Trennung, erhöht die Vergleichbarkeit und erlaubt belastbare Aussagen über die Wirkung einzelner Schritte und deren Zusammenspiel. Auf dieser methodischen Grundlage

wird in den folgenden Abschnitten der genaue Aufbau und Ablauf des Experimentierplans beschrieben. Dabei wird zunächst auf die isolierte Anwendung einzelner Verfahren eingegangen, bevor im Anschluss kombinierte Pipelines betrachtet werden.

5.3.2 Untersuchung von Einzelverfahren

Bevor einzelne Verfahren zur Datenvorverarbeitung isoliert untersucht werden, erfolgt zunächst die Analyse einer Baseline. Dabei wird das Analysemodell ohne jegliche Datenvorverarbeitung direkt auf die Rohdaten angewendet. Diese Referenzkonfiguration dient als methodischer Ausgangspunkt, an dem sich sämtliche Veränderung durch die nachfolgenden Verfahrensschritte relativ messen lassen. Die Wahl eines konstanten Klassifikationsmodells mit festen Parametern stellt sicher, dass die Ergebnisse der Baseline valide und konsistent sind. Dabei wird berücksichtigt, dass die Rohdaten Verzerrungen enthalten können. Die Baseline bildet somit eine praxisnahe und unverfälschte Ausgangssituation ab.

Im Anschluss an die Baseline erfolgt die Analyse der Einzelverfahren, da diese eine methodisch fundierte Referenz für die folgenden Verfahrenskombinationen darstellen. Alle ausgewählten DVV, sowohl zur Behandlung fehlender Werte als auch zur Reduktion redundanter Merkmale und Instanzen, werden jeweils einzeln auf den Rohdatensatz angewendet. Nach jeder Anwendung erfolgt die Bewertung der Auswirkungen anhand des Data-Mining-Verfahrens. Dieses Vorgehen erlaubt eine differenzierte Einschätzung der isolierten Wirkung jedes Verfahrens im Hinblick auf Modellgüte, Effizienz und Robustheit. Die Anwendung einzelner DVV stellt eine methodisch notwendige Grundlage dar, um die spezifische Wirkung jedes Verfahrens nachvollziehen zu können. Sie ermöglicht es, die Veränderung der Modellperformance eindeutig auf die jeweilige Maßnahme zurückzuführen und so deren Effekt empirisch abzusichern. Insbesondere bei Verfahren zur Behandlung fehlender Werte und zur Datenreduktion kann die Wirkung einzelner Verfahren sehr unterschiedlich ausfallen, wie in den Abschnitten 0 und 4.4 deutlich wurde. Ein Imputationsverfahren kann bspw. systematische Verzerrungen verursachen oder bestimmte Verteilungen glätten, während eine Merkmalsselektion den Informationsgehalt reduzieren oder die Interpretierbarkeit erhöhen kann. Ohne eine isolierte Betrachtung besteht das Risiko, dass solche Effekte überlagert oder falsch interpretiert werden, wenn mehrere Verfahren in einer Pipeline eingesetzt werden. Darüber hinaus fungieren die Ergebnisse der Einzelanalysen als notwendige Referenzwerte für die spätere Bewertung von Verfahrenskombinationen. Nur wenn bekannt ist, wie sich ein Verfahren allein auf das Modell auswirkt, lässt sich bewerten, ob die Kombination mit einem weiteren Verfahren zu einer Verbesserung, Neutralisierung oder Verschlechterung der Analysequalität führt. Diese Differenzierbarkeit bildet die Grundlage für alle nachgelagerten Vergleiche im Experimentierplan. Die Durchführung erfolgt standardisiert, jedes Verfahren wird also nach einem einheitlichen Ablauf angewendet und der Effekt anschließend im selben Data-Mining-Verfahren mithilfe der definierten Evaluationsmetriken bewertet. So ist sichergestellt, dass die Resultate unter kontrollierten Bedingungen zustande kommen und sich ausschließlich auf das jeweilige DVV zurückführen lassen. Die Ergebnisse der isolierten Anwendung der Verfahren liefern die notwendige Referenz, um die Wirkung komplexerer Konfigurationen einordnen zu können. Aufbauend darauf wird nachfolgend untersucht, wie sich mehrere DVV unterschiedlicher Ursachenbehandlung in Kombination verhalten, wobei insbesondere Interaktionen und Reihenfolgeeffekte differenziert analysiert werden. Die nachfolgende Abbildung 5.1 zeigt den schematischen Aufbau der experimentellen Konfiguration zur Analyse einzelner DVV.

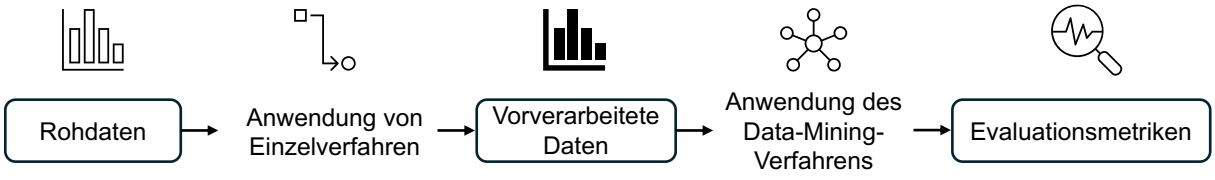


Abbildung 5.1: Schematische Darstellung der Anwendung von Einzelverfahren im Experimentierplan

Abbildung 5.1 visualisiert den Versuchsaufbau im Experimentierplan für die Anwendung von Einzelverfahren. Dabei zeigen eingerahmte Elemente jeweils das Ergebnis aus den nicht eingerahmten Elementen, den Aktionen. Ausgehend von denselben Rohdaten wird jeweils ein einzelnes DVV auf den Datensatz angewendet. Im Anschluss erfolgt die Anwendung des konstanten Analysemodells und die Bewertung anhand der zuvor definierten Metriken. Diese Konfiguration ermöglicht eine direkte Zuordnung beobachteter Effekte zur jeweiligen Maßnahme. Die Untersuchung der Einzelverfahren bildet einen zentralen Bestandteil der Versuchsarchitektur, da sie eine klare Trennung und Bewertung der isolierten Effekte ermöglicht. Aufbauend auf diesem Konzept wird im nächsten Schritt der Experimentierplan um Kombinationen von DVV erweitert, um auch Interaktionen und Reihenfolgeeffekte systematisch berücksichtigen zu können.

5.3.3 Untersuchung von Verfahrenskombinationen und Reihenfolgevariation

Ausgehend von den zuvor gewonnenen Einzelanalysen richtet sich der Fokus nun auf die kombinierte Anwendung mehrerer DVV. Dabei werden jeweils ein Verfahren zur Behandlung fehlender Werte und ein Verfahren zur Datenreduktion nacheinander in der Versuchspipeline auf den Rohdatensatz angewendet. Unter dem Begriff Datenreduktion werden sowohl Merkmalsreduktion als auch Instanzenreduktion zusammengefasst. Die beiden Kategorien unterscheiden sich hinsichtlich ihrer Eingriffsrichtung, haben dennoch beide eine Verringerung des Umfangs der Eingangsdaten als Ziel. Daher werden sie im Rahmen des Experimentierplans als Verfahren zur Datenreduktion zusammengefasst und nicht kategorisch unterschieden.

Für jede Kombination werden jeweils zwei Verfahren auf die Daten angewendet, wobei die Reihenfolge variiert wird. Jede Kombination wird in zwei Varianten getestet, wobei einmal die Datenreduktion vor der Behandlung fehlender Werte und einmal die Behandlung fehlender Werte vor der Datenreduktion angewendet wird. Diese Reihenfolgevariation ist erforderlich, da die Anwendung eines Verfahrens die Ausgangslage für das jeweils nachfolgende Verfahren verändert. Wird bspw. zuerst ein Imputationsverfahren durchgeführt, fließen ergänzte Werte in die nachfolgende Merkmals- oder Instanzenreduktion ein, was die Auswahl relevanter Strukturen beeinflussen kann. Umgekehrt verändert ein vorgelagertes Datenreduktionsverfahren den Informationsgehalt der Daten, was wiederum dazu führen kann, dass wichtige Informationen für die Imputation nicht mehr zur Verfügung stehen und die Schätzgenauigkeit sinkt. In beiden Fällen entsteht eine veränderte Ausgangslage, die die Wirkung der Gesamtverarbeitung beeinflusst. Nur durch den Vergleich beider Reihenfolgen lassen sich solche Interaktionseffekte identifizieren und bewerten.

Nach jeder durchlaufenen Pipeline, ob Einzelverfahren oder Kombination, wird das standardisierte Data-Mining-Verfahren erneut trainiert und bewertet. Zwischenmessungen innerhalb der Pipelines entfallen, da alle betrachteten Verfahren bereits in isolierter Form analysiert werden. Diese methodische Trennung stellt sicher, dass beobachtete Effekte eindeutig auf bestimmte Verfahrensschritte oder Kombinationen zurückgeführt werden können. Nach jeder vollständigen Vorverarbeitungskette wird das Data-Mining-Verfahren auf die transformierten Daten angewendet, wobei die Bewertung über die zuvor ausgewählten Evaluationsmetriken erfolgt. Dadurch wird gewährleistet, dass jede Konfiguration unter identischen Bewertungsbedingungen analysiert wird und sowohl unmittelbare als auch übergreifende Effekte systematisch vergleichbar bleiben. Die gesamte Durchführung erfolgt unter konstanten Rahmendbedingungen, sodass externe Einflussfaktoren ausgeschlossen werden können und die Ergebnisse vergleichbar bleiben. In der nachfolgenden Abbildung 5.2 ist die Vorgehensweise im Experimentierplan mit Verfahrenskombinationen schematisch dargestellt.

Abbildung 5.2 zeigt den Versuchsaufbau im Experimentierplan für Verfahrenskombinationen in zwei Durchläufen. Im ersten Durchlauf erfolgt zunächst die Behandlung fehlender Werte (DVV A), gefolgt von der Datenreduktion (DVV B). Im zweiten Durchlauf wird die Reihenfolge der Verfahrensanwendung umgekehrt und so zunächst die Datenreduktion (DVV B) und anschließend die Imputation (DVV A) vorgenommen. In beiden Fällen wird nach Abschluss der vollständigen Datenvorverarbeitung das standardisierte Analysemodell auf die transformierten

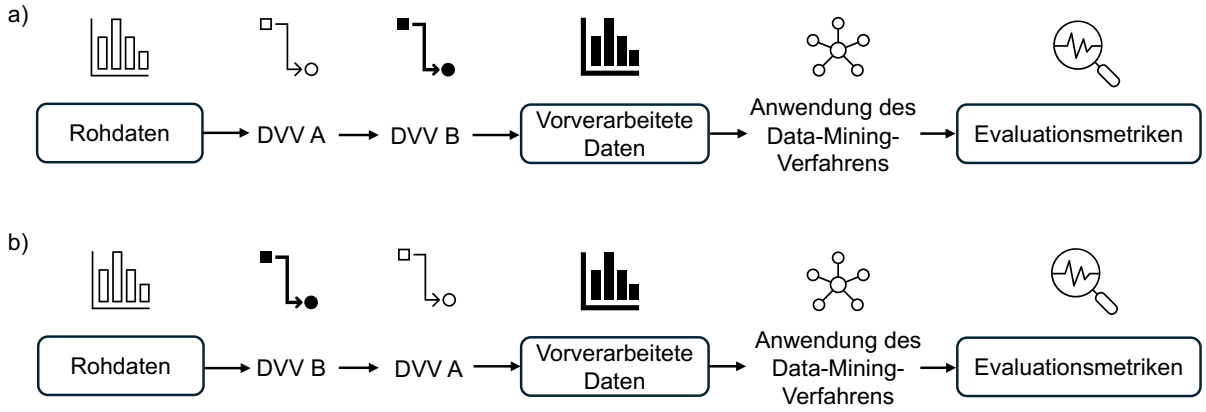


Abbildung 5.2: Schematische Darstellung der Verfahrenskombination im Experimentierplan

Daten angewendet. Dieses Vorgehen erlaubt es, systematisch zu untersuchen, ob und in welcher Weise die Reihenfolge der DVV die Analyseergebnisse beeinflusst. Die Konzeption des Experimentierplans bildet damit die methodische Grundlage für die nachfolgende empirische Untersuchung. Im folgenden Kapitel wird diese Konzeption in eine konkrete Implementierung überführt, um Verfahren zur Behandlung fehlender Werte und zur Datenreduktion unter kontrollierten Bedingungen systematisch zu testen und ihre Wirkung auf ein nachgelagertes Data-Mining-Verfahren zu messen.

6 Praktische Umsetzung des Experimentierplans

Die Umsetzung des im vorherigen Abschnitt entwickelten Experimentierplans erfolgt im Rahmen einer modular aufgebauten Implementierung, die eine systematische Untersuchung der Wirkung von DVV ermöglicht. Ziel ist es, das in Abschnitt 5.3 beschriebene Versuchsdesign mit realen Produktionsdaten nachzubilden und die Auswirkungen einzelner sowie kombinierter DVV mithilfe der in Abschnitt 5.1.4 ausgewählten Evaluationsmetriken anhand eines nachgelagerten Data-Mining-Verfahrens zu bewerten. Die Implementierung erfolgt in der Programmiersprache Python, da diese eine hohe Flexibilität bei der Verarbeitung, Transformation und Analyse von Daten bildet. Zum Einsatz kommen dabei unter anderem die Bibliotheken `pandas` (Datenmanagement und Vorverarbeitung) (pandas development team, 2025), `NumPy` (numerische Operationen) (NumPy developers, 2023) und `scikit-learn` (Imputation, Merkmalsselektion, Modelltraining und Evaluation). Die gesamte Analyse wird in einer Jupyter-Notebook-Umgebung durchgeführt, um eine strukturierte, dokumentierbare und reproduzierbare Umsetzung der einzelnen Schritte zu ermöglichen. Die Durchführung aller Versuche erfolgt lokal auf einem Apple MacBook Air (Modell MacBookAir10,1) mit Apple M1 Chip und 8 GB Arbeitsspeicher. Das Gerät verfügt über acht CPU-Kerne (4 Performance- und 4 Effizienzkerne) und arbeitete unter macOS 15.6 (Kernel-Version Darwin 24.6.0). Als Laufzeitumgebung diente Python 3.11.8 in Verbindung mit einer Jupyter-Notebook-Oberfläche. Alle verwendeten Bibliotheken wurden auf aktuelle Versionen abgestimmt, um eine konsistente Ausführungsumgebung und damit die Reproduzierbarkeit der Experimente zu gewährleisten. Die in den Experimenten eingesetzten Bibliotheken samt Versionsangaben sind in Anhang E dokumentiert. Für die Umsetzung des Experimentierplans wird ein geeigneter Datensatz benötigt, der die Durchführung der vorgesehenen Schritte ermöglicht. Ausgehend von den methodischen Zielen ergeben sich dabei konkrete Anforderungen an Struktur und Inhalt der verwendeten Daten.

6.1 Datengrundlage und Beschreibung

Für die Umsetzung des experimentellen Designs wird ein Datensatz benötigt, der bestimmte strukturelle und inhaltlich Anforderungen erfüllt. Diese Anforderungen ergeben sich direkt aus dem Ziel des Experimentierplans, verschiedene DVV zur Behandlung fehlender Werte und zur Datenreduktion hinsichtlich ihrer Wirkung auf die Analyseleistung des nachgelagerten Data-Mining-Verfahrens systematisch zu bewerten. Eine wesentliche Voraussetzung ist das Vorhandensein fehlender Werte, da nur so unterschiedliche Verfahren zu deren Behebung hinsichtlich ihrer Wirkung untersucht und verglichen werden können. Darüber hinaus wird eine ausreichend hohe Anzahl von Merkmalen vorausgesetzt, um Verfahren zur Merkmals- und Instanzenreduktion angemessen anwenden können. Die Merkmale sollten ein gewisses Maß an Redundanz oder Korrelation aufweisen, damit Verfahren zur Merkmalsreduktion auf reale Zusammenhänge reagieren und mögliche Auswirkungen wie Informationsverlust oder Modellvereinfachung überhaupt messbar werden. Eine binäre Zielvariable ist erforderlich, um die Auswirkungen der Datenvorverarbeitung im Rahmen eines Klassifikationsmodells quantitativ anhand der ausgewählten Evaluationsmetriken bewerten zu können. Zusätzlich sollte der Datensatz in einem möglichst unbereinigten Zustand vorliegen, damit die Wirkung der Anwendung einzelner DVV effektiv und nachvollziehbar ist.

Ausgehend von diesen Anforderungen wird im Folgenden der Datensatz vorgestellt, der als Grundlage für die experimentelle Umsetzung dient. Bei dem eingesetzten Datensatz handelt es sich um reale Produktionsdaten aus der Halbleiterfertigung, die im Rahmen eines industriellen Qualitätssicherungsprozesses erhoben wurden. Der Datensatz hat den Titel SECOM (Semi-Conductor Manufacturing) und wurde über das UC Irvine Machine Learning Repository bereitgestellt (McCann & Johnston, 2008). Jede Beobachtung im Datensatz repräsentiert eine Produktionseinheit, für die 590 Prozessvariablen über Sensoren oder Messpunkte aufgezeichnet wurden. Die einzelnen Merkmale im SECOM-Datensatz sind nicht näher spezifiziert, es handelt sich um anonymisierte numerische Prozessgrößen. Diese werden ergänzt durch einen Zeitstempel, der den Zeitpunkt der Qualitätskontrolle angibt, sowie eine Zielvariable, die das

Ergebnis der Prüfung kodiert. Der Zielwert ist binär ausgeprägt, und nimmt einen Wert von -1 für bestandene Bauteile sowie einen Wert von 1 für fehlerhafte Bauteile an. Die folgende Abbildung 6.1 veranschaulicht die grundlegende Struktur des eingelesenen Datensatzes sowie zentrale Kennzahlen zur Zielvariablenverteilung und Datenqualität.

```
DataFrame-Form: (1567, 592)
   label            timestamp        0        1          2  ...       589
0     -1  19/07/2008 11:55:00  3030.93  2564.00  2187.7333  ...       NaN
1     -1  19/07/2008 12:32:00  3095.78  2465.14  2230.4222  ...  208.2045
2      1  19/07/2008 13:17:00  2932.61  2559.94  2186.4111  ...   82.8602

Anteil fehlender Werte: 4.52% (41951 von 927664 Werten)

Anzahl numerischer Merkmale: 590
Nahezu konstante Merkmale: 116 von 590
Vollständig fehlende Merkmale: 0

Verteilung der Zielvariable:
   Klasse  Anzahl  Anteil (%)
0      -1    1463       93.36
1       1     104        6.64
```

Abbildung 6.1: Struktur und zentrale Kennzahlen des SECOM-Datensatzes (Python-Output)

Die Abbildung 6.1 zeigt einen Python-Output, der eine Übersicht über die Struktur und Beschaffenheit der eingelesenen Daten liefert. Der obere Teil der Abbildung veranschaulicht anhand der Darstellung der ersten Zeilen, wie die einzelnen Beobachtungen im Datensatz strukturiert sind. Jede Zeile umfasst demnach eine Zielvariable, einen Zeitstempel sowie 590 numerische Prozessmerkmale, wobei bereits ein Wert als fehlend angezeigt wird (Zeile 0, Spalte 589). Das Fehlen des Wertes wird mit NaN gekennzeichnet. Unter den Beobachtungen folgen zusammenfassende Kennzahlen zur Datenqualität. Der Datensatz weist insgesamt einen Anteil fehlender Werte von 4,52 % auf, was einer absoluten Anzahl von 41951 entspricht. Von den 590 Merkmalen gelten 116 als nahezu konstant, was bedeutet, dass sie über alle Instanzen hinweg nur eine minimale Streuung zeigen. Diese Einschätzung basiert auf der Berechnung der Standardabweichung je Merkmal. Merkmale mit einem Streuungswert von weniger als 0,00001 wurden dabei als nahezu konstant klassifiziert. Gleichzeitig existieren keine vollständig fehlenden Merkmale. Jede Spalte enthält somit mindestens einen gültigen Wert. Abschließend zeigt die Abbildung die Verteilung der Zielvariable. Demnach entfallen 93,36 % der Instanzen auf die Kategorie −1 (bestanden), während nur 6,64 % als fehlerhaft klassifiziert sind. Diese Unausgeglichenheit ist typisch für industrielle Fertigungsprozesse, in denen eine geringe Fehlerrate angestrebt wird. Gleichzeitig stellt sie eine Herausforderung für die Modellbildung dar, da viele Klassifikationsverfahren dazu tendieren, die dominante Klasse zu bevorzugen und seltener auftretende Fehlerfälle zu übersehen, wie in Abschnitt 2.2.2 beschrieben. Umso wichtiger ist eine differenzierte Bewertung des Modells im Hinblick auf Robustheit und Klassifikationsgüte.

Um das Auftreten fehlender Werte differenzierter zu betrachten, folgt ergänzend eine grafische Darstellung in Form der in Abschnitt 3.2 eingeführten Missingness Matrix. Während der Anteil fehlender Werte einen allgemeinen Überblick bietet, liefert die Visualisierung Hinweise auf mögliche Muster in der Verteilung der Leerstellen. So lässt sich erkennen, ob bestimmte Merkmale besonders häufig betroffen sind oder ob sich das Fehlen auf einzelne Gruppen von Instanzen konzentriert. Diese Einblicke sind besonders relevant, da die Struktur der fehlenden Werte beeinflusst, welche Verfahren zur Behandlung der fehlenden Werte im weiteren Verlauf geeignet sind. Abbildung 6.2 zeigt einen Ausschnitt der Missingness Matrix des SECOM-Datensatzes. Dargestellt sind aufgrund des hohen Umfangs des gesamten Datensatzes 250 zufällig ausgewählte Instanzen sowie das erste Drittel der 590 Merkmale.

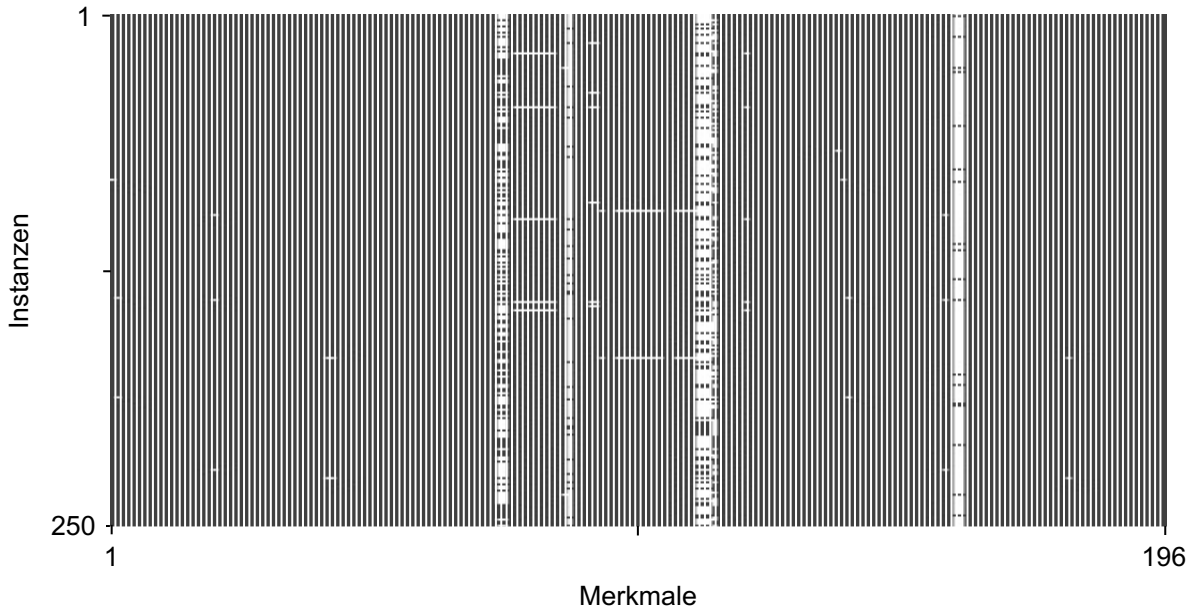


Abbildung 6.2: Missingness-Matrix des SECOM-Datensatzes (Ausschnitt)

In Abbildung 6.2 entspricht jede horizontale Linie einem Merkmal, das über die ausgewählten Instanzen hinweg dargestellt wird. Die Linien sind durchgängig, wenn für ein Merkmal bei einer Instanz ein Wert vorliegt und zeigen Unterbrechungen an den Stellen, an denen Werte fehlen. Diese Darstellung macht auf einen Blick sichtbar, bei welchen Merkmalen und Instanzen Datenlücken auftreten und ob sich dabei systematische Muster erkennen lassen. Die Abbildung verdeutlicht, dass die fehlenden Werte nicht gleichmäßig über alle Merkmale verteilt sind, sondern sich auf bestimmte Spalten konzentrieren. Dieses Muster legt nahe, dass systematische Ursachen für das Fehlen der Daten vorliegen könnten, bspw. defekte Sensoren oder unterbrochene Messphasen. Die Visualisierung bestätigt damit die zuvor beschriebene Menge fehlender Werte und liefert gleichzeitig erste Hinweise auf mögliche Strukturen im Auftreten der Lücken, die bei der Auswahl geeigneter Imputationsverfahren berücksichtigt werden können.

Zusammenfassend lässt sich festhalten, dass sich der SECOM-Datensatz eine geeignete Grundlage für die Umsetzung des entwickelten Experimentierplans darstellt. Er erfüllt grundlegende Anforderungen wie das Vorhandensein fehlender Werte, eine ausreichende Anzahl an Merkmalen und Instanzen sowie eine binäre Zielvariable. Zudem stammen die Daten aus einem realen industriellen Fertigungsprozess, was typische Eigenschaften praktischer Produktionsumgebungen abbildet, wie etwa das musterartige Fehlen von Werten infolge von bspw. Sensordefekten. Auch wenn die Merkmale nicht näher beschrieben sind und damit keine inhaltliche Interpretation einzelner Variablen möglich ist, stellt dies für die angestrebte Analyse von DVV keine wesentliche Einschränkung dar. Die stark unausgeglichene Klassenverteilung der Zielvariable hingegen ist methodisch relevant und wird bei der Modellierung und Bewertung entsprechend berücksichtigt. Auf Grundlage dieser Datensatzeigenschaften sowie der methodischen Zielsetzung wird im nächsten Schritt ein Klassifikationsmodell festgelegt, welches in allen Experimenten konsistent eingesetzt wird. Ziel ist es, zunächst eine Baseline-Leistung auf dem unvorbereiteten Datensatz zu ermitteln, um darauf aufbauend die Wirkung einzelner und kombinierter DVV analysieren zu können.

6.2 Auswahl des Klassifikationsmodells und Baseline-Versuch

Für die experimentelle Analyse wird ein einheitliches Klassifikationsmodell definiert, das in sämtlichen Versuchskonfigurationen verwendet wird. Durch die einheitliche Modellwahl lassen sich Unterschiede in den Evaluationsmetriken eindeutig auf die jeweils angewandten DVV zurückführen. Da der SECOM-Datensatz fehlende Werte enthält, muss ein Klassifikator ausgewählt werden, der mit fehlenden Werten umgehen kann. Der in Abschnitt 2.2.1 beschriebene

verbreitete `RandomForestClassifier` aus der Python-Bibliothek `scikit-learn` kann nicht mit fehlenden Werten umgehen, was ihn in diesem Anwendungskontext disqualifiziert. Stattdessen kommt im vorliegenden Experimentierplan der `HistGradientBoostingClassifier` aus `scikit-learn` zum Einsatz, der auf dem in Abschnitt 2.2.1 beschriebenen Histogramm-basierten Gradient Boosting beruht. Dieses Modell kann fehlende Werte verarbeiten und ist für numerische sowie hochdimensionale Daten geeignet. Es benötigt keine vorherige Skalierung der Merkmale und kann auch mit nichtlinear verteilten Eingabedaten robust umgehen. Zudem zeichnet sich das Modell durch eine hohe Effizienz in der Verarbeitung größerer Datensätze aus, da es Histogramme zur Approximation der Merkmalsverteilungen verwendet.

Zur Bestimmung der Baseline-Leistung wird der Klassifikator zunächst auf den unvorbereiteten SECOM-Rohdatensatz angewendet. Die resultierenden Evaluationsmetriken dienen als Referenz für die spätere Bewertung der Wirkung einzelner Verfahren und Verfahrenskombinationen. Da die Vielzahl der geplanten Experimente mit dem hochdimensionalen SECOM-Datensatz einen hohen Rechenaufwand verursacht, wurde auf eine zehnfache Kreuzvalidierung verzichtet. Wie bereits in Abschnitt 2.2.2 erläutert, existiert kein fixes Optimum für die Anzahl der Folds. Bereits eine geringere Zahl an Folds kann stabile und aussagekräftige Ergebnisse liefern und zugleich die Rechenzeit erheblich reduzieren. Vor diesem Hintergrund wird in dieser Arbeit eine dreifache Kreuzvalidierung eingesetzt, bei der der Datensatz jeweils in drei gleich große Teilmengen unterteilt. In jedem Durchlauf wird das Modell auf zwei dieser Teilmengen trainiert und auf der verbleibenden Teilmenge getestet. Die so erzielten Ergebnisse werden gemittelt, um eine robuste und verlässliche Einschätzung der Modellgüte zu erhalten. Zur quantitativen Bewertung werden die in Abschnitt 5.1.4 ausgewählten Metriken Accuracy, Precision, Recall und die FM herangezogen, welche in Anlehnung an die Implementierung in `scikit-learn` im Folgenden als F1-Score bezeichnet wird. Darüber hinaus werden die Rechenzeit zur Bewertung der Effizienz sowie die Standardabweichung der Validierungsergebnisse zur Bewertung der Robustheit berücksichtigt. Die Ausgabe in Abbildung 6.3 zeigt die Evaluationsmetriken des Baseline-Versuchs auf dem unvorbereiteten SECOM-Datensatz.

```
Baseline (ohne Vorverarbeitung)
------------------------------------------------------------
Accuracy (MW ± SD):              0.8342 ± 0.1410
Precision (MW ± SD):             0.0405 ± 0.0572
Recall (MW ± SD):                0.2381 ± 0.3367
F1-Score (MW ± SD):              0.0692 ± 0.0978
Rechenzeit (Sekunden):                     9.92
Anzahl Merkmale:                            590
Anzahl Instanzen:                          1567
Verteilung Zielvariable:      {-1: 1463, 1: 104}
```

Abbildung 6.3: Ergebnisse des Baseline-Versuchs auf dem unvorbereiteten Rohdatensatz

Die Ergebnisse des Baseline-Versuchs in Abbildung 6.3 zeigt eine durchschnittliche Accuracy von 0,8342 bei einer Standardabweichung von ± 0,1410. Damit erzielt das Modell auf den ersten Eindruck eine hohe Gesamtgenauigkeit. Bei Betrachtung der weiteren Kennzahlen wird jedoch deutlich, dass das Modell fehlerhafte Bauteile kaum erkennt. Die Precision beträgt 0,0405 (± 0,0572) und zeigt damit, dass lediglich etwa vier Prozent der vom Modell als defekt klassifizierten Instanzen tatsächlich fehlerhaft sind. Der Recall liegt bei 0,2381 (± 0,3367), sodass nur knapp ein Viertel der tatsächlich defekten Bauteile erkannt wurde. Die hohe Standardabweichung dieses Wertes unterstreicht zudem eine starke Instabilität über die Kreuzvalidierungsdurchläufe. Der daraus resultierende F1-Score von 0,0692 (± 0,0978) bestätigt, dass weder eine verlässliche Erkennungsrate noch eine ausreichende Präzision bei der Vorhersage defekter Instanzen erzielt werden konnte. Insgesamt lässt sich festhalten, dass das Modell in dieser Konfiguration keine praktikable Lösung für die vorliegende Klassifikationsaufgabe darstellt.

Die Ursache liegt in der stark unausgeglichenen Zielverteilung des SECOM-Datensatzes, in dem nur 6,64 % der Instanzen als fehlerhaft klassifiziert sind. Wie in Abschnitt 2.2.2 beschrieben, neigen Lernalgorithmen in solchen Fällen dazu, die Minderheitsklasse zu vernachlässigen. Aus diesem Grund wird im nächsten Schritt das in Abschnitt 2.3 beschriebene SMOTE-Verfahren eingesetzt, bei der es sich um ein Verfahren zur gezielten Generierung synthetischer Instanzen der Minderheitsklasse handelt, um eine ausgewogene Verteilung der Zielvariable herzustellen. Auf dieser Grundlage wird das Modell in die Lage versetzt, auch seltener auftretende Fehlerfälle zu lernen und zu klassifizieren. Ziel ist es, die Erkennungsleistung zu verbessern und damit eine praxisrelevante Prognosefähigkeit zu erreichen. Die Integration von SMOTE bildet zudem eine methodisch notwendige Voraussetzung für den nachgelagerten Experimentierplan, da die Wirkung verschiedener DVV nur dann vergleichbar untersucht werden kann, wenn das zugrunde liegende Klassifikationsmodell in der Lage ist, beide Klassen prinzipiell zu unterscheiden. In der vorliegenden Implementierung wird SMOTE über die Python-Bibliothek `imbalanced-learn` (imbalanced-learn developers, 2023) angewendet. Die Ausgabe in Abbildung 6.4 zeigt die Ergebnisse des angepassten Baseline-Versuchs, in dem SMOTE zur Ausbalancierung der Zielvariable angewendet wurde.

```
Dummy-Imputation + SMOTE → Dummys zurück zu NaN
------------------------------------------------------------
Accuracy (MW ± SD):              0.8719 ± 0.0995
Precision (MW ± SD):             0.8211 ± 0.1314
Recall (MW ± SD):                0.9850 ± 0.0213
F1-Score (MW ± SD):              0.8911 ± 0.0832
Rechenzeit (Sekunden):                    11.75
Anzahl Merkmale:                            590
Anzahl Instanzen:                          2926
Verteilung Zielvariable:     {-1: 1463, 1: 1463}
```

Abbildung 6.4: Ergebnisse des Baseline-Versuchs mit SMOTE

Der angepasste Baseline-Versuch mit SMOTE zeigt in der Ausgabe in Abbildung 6.4 verglichen mit der ursprünglichen Konfiguration eine deutlich verbesserte Modellgüte. Während im ersten Versuch kaum fehlerhafte Instanzen erkannt wurden (Recall = 0,2381), identifiziert das Modell nach Anwendung von SMOTE über 98 % der Fehler korrekt (Recall = 0,9850) bei gleichzeitig hoher Precision (0,8211). Der F1-Score steigt entsprechend auf 0,8911 an. Neben den Mittelwerten der Modellgüte-Metriken wird erneut auch deren Standardabweichung über die drei Kreuzvalidierungsdurchläufe zur Bewertung der Robustheit herangezogen. Im Gegensatz zur ursprünglichen Baseline-Konfiguration zeigen sich hier nachvollziehbare Streuungen, die eine inhaltlich interpretierbare Aussage zur Stabilität des Modells ermöglichen. Die relativ geringen Standardabweichungen bei Accuracy (0,0995), Precision (0,1314), Recall (0,0213) und F1-Score (0,0832) deuten auf eine konsistente Modellleistung hin, die nur in geringem Maße durch Variation in den Teilmengen des Datensatzes beeinflusst wird. Diese Werte sprechen für eine robuste Klassifikation unter den gegebenen Bedingungen. Die Kombination aus hoher Erkennungsleistung und niedriger Streuung bestätigt, dass die Anwendung von SMOTE nicht nur zu einer verbesserten, sondern auch zu einer stabileren Modellgüte führt. Damit bestätigt sich, dass die Ausbalancierung der Zielvariable mittels SMOTE eine grundlegende Voraussetzung für eine verlässliche Bewertung der Modellleistung und damit auch der Wirkung von DVV ist. Auf dieser Grundlage können nun verschiedene Verfahren der Datenvorverarbeitung gezielt und vergleichbar untersucht werden. Im folgenden Abschnitt wird dafür zunächst festgelegt, welche konkreten Verfahren ausgewählt und im Experimentierplan berücksichtigt werden.

6.3 Auswahl der Datenvorverarbeitungsverfahren

Die Auswahl geeigneter Verfahren erfolgt mit Blick auf ihre praktische Einsetzbarkeit im Rahmen des entwickelten Experimentierplans. Die Auswahl orientiert sich an den in Abschnitt 5.2

definierten Kriterien, insbesondere an der Eignung für reale industrielle Datenprobleme, der Modularität, der Rechenökonomie, der technischen Verfügbarkeit sowie der methodischen Interpretierbarkeit. Grundlage für die Auswahl bieten die in den Abschnitten 3.3 und 4.2 bzw. 4.3 beschriebenen Verfahren.

6.3.1 Auswahl der Verfahren zur Behandlung fehlender Werte

Für die experimentelle Analyse der Auswirkungen fehlender Werte auf die Modellleistung werden fünf Imputationsverfahren ausgewählt. Berücksichtigt werden zunächst drei statistische Verfahren, die sich trotz methodischer Einschränkungen durch eine hohe Effizienz, breite Verfügbarkeit und einfache Integration in den Experimentierplan auszeichnen. Die Mittelwertimputation ersetzt fehlende Werte durch den arithmetischen Mittelwert eines Merkmals (vgl. Abschnitt 3.3.3). Sie ist leicht interpretierbar, rechenökonomisch und über `sklearn.impute.SimpleImputer` in der Bibliothek `scikit-learn` unkompliziert in Python umsetzbar. Obwohl sie die Varianzstruktur verzerren kann, erlaubt sie eine erste Einschätzung der Modellrobustheit durch einfache Ersetzungen. Ergänzend wird die Medianimputation berücksichtigt, bei der fehlende Werte durch den Median eines Merkmals ersetzt werden. Dieses Verfahren ist besonders für numerische Daten geeignet, da es robust gegenüber Ausreißern ist und auch bei schief verteilten Variablen eine verlässlichere Schätzung als der arithmetische Mittelwert liefert (vgl. Abschnitt 3.3.3). Die Umsetzung erfolgt ebenfalls über den `SimpleImputer` in `scikit-learn`, wobei lediglich die Strategie entsprechend angepasst wird. Die Konstantenimputation schließlich ersetzt alle fehlenden Werte durch einen festen Wert, wie z. B. 0 (vgl. Abschnitt 3.3.3). Auch dieses Verfahren ist über den `SimpleImputer` direkt implementierbar. Es erfüllt zwar nicht die Anforderungen an Interpretierbarkeit und statistische Validität, wird jedoch gezielt eingesetzt, um die Auswirkungen trivialer Imputationsverfahren im Vergleich zu fundierteren Verfahren zu prüfen. Zusammen bieten diese drei Verfahren eine Referenz für die Bewertung robusterer Imputationsverfahren. Ausschlaggebend für die Entscheidung, diese drei einfachen Verfahren einzubeziehen, ist neben ihrer unkomplizierten Umsetzbarkeit der geringe zu erwartende Rechenaufwand, der eine effiziente Durchführung der Experimente ermöglicht.

Als multivariate Verfahren mit höherer Aussagekraft werden die kNN-Imputation und die iterative Imputation (vgl. Abschnitt 3.3.5) in die Analyse einbezogen. Die kNN-Imputation berücksichtigt lokale Ähnlichkeiten im Merkmalsraum und eignet sich insbesondere für strukturierte numerische Daten. Die Implementierung erfolgt über `sklearn.impute.KNNImputer`, wodurch eine nahtlose Integration in Pipelines möglich ist. Trotz Einschränkungen in hochdimensionalen Räumen ist das Verfahren modular einsetzbar, nachvollziehbar und stabil verfügbar. Die iterative Imputation basiert auf sequenziellen Regressionsmodellen und wird in `scikit-learn` durch `sklearn.impute.IterativeImputer` unterstützt. Sie berücksichtigt multivariate Zusammenhänge und ist insbesondere dann geeignet, wenn eine realitätsnahe Rekonstruktion der Datenstruktur angestrebt wird. Beide Verfahren gelten als robust gegenüber Datenvariationen und erfüllen grundlegende Kriterien wie Modularität, methodische Interpretierbarkeit und technische Verfügbarkeit.

Neben den statistischen und multivariaten Verfahren wurde auch geprüft, modellbasierte Verfahren in die Analyse einzubeziehen. Diese gelten zwar als besonders leistungsfähig bei komplexen Interaktionen und nichtlinearen Zusammenhängen (vgl. Abschnitt 3.3.6), erwiesen sich in den Tests jedoch als mit einem erheblich höheren Rechenaufwand verbunden. Da Effizienz ein zentrales Auswahlkriterium in der Methodik darstellt und sämtliche Verfahren in alle Versuchskonfigurationen integriert werden müssen, wurde auf den Einsatz modellbasierter Verfahren verzichtet. Auf den Einsatz von Eliminierungsverfahren wird ebenfalls verzichtet, da sie bei einem relevanten Anteil fehlender Werte zu erheblichen Informationsverlusten führen und die Aussagekraft der Ergebnisse verzerren können (vgl. Abschnitt 3.3.1). Ebenso werden klassenbasierte Verfahren nicht berücksichtigt, da ihre Bildung homogener Gruppen bei überwiegend numerischen und hochdimensionalen Daten kaum praktikabel ist und sie zudem multivariate Abhängigkeiten zwischen Variablen nur unzureichend berücksichtigen können (vgl. Abschnitt 3.3.4).

Insgesamt deckt die Auswahl ein breites Spektrum von Verfahren zur Behandlung fehlender Werte ab, von einfachen statistischen bis hin zu komplexeren iterativen Verfahren. Sie erfüllt damit grundlegende Anforderungen an die methodische Fundierung, Vergleichbarkeit und praktische Anwendbarkeit im Rahmen des Experimentierplans und bildet eine belastbare Grundlage für die nachfolgenden Analyseschritte. Die folgende Tabelle 6.1 gibt einen Überblick über die ausgewählten Verfahren und bewertet sie anhand der Auswahlkriterien.

Tabelle 6.1: Bewertung der ausgewählten Imputationsverfahren anhand methodischer Kriterien

Verfahren	**Eignung für Realdaten**	**Robustheit**	**Modularität & Kombinierbarkeit**	**Rechenökonomie**	**Interpretierbarkeit**	**Verfügbarkeit**
Mittelwertimputation	◑	○	●	●	●	●
Medianimputation	◑	◑	●	●	●	●
Konstantenimputation	◑	○	●	●	○	●
kNN-Imputation	●	◑	●	◑	◑	●
Iterative Imputation	●	◑	●	○	◑	●

Die Tabelle 6.1 veranschaulicht die Bewertung der eingesetzten Imputationsverfahren anhand zentraler Kriterien, wobei volle, halbe und leere Kreise eine gute, teilweise oder geringe Erfüllung der Auswahlkriterien beschreiben. Die Mittelwertimputation wird insgesamt als positiv bewertet, hat jedoch Einschränkungen in Bezug auf Eignung für Realdaten und Robustheit.

Die Mittelwertimputation führt durch das Ersetzen aller fehlenden Werte mit demselben Wert zu einem Verlust an Varianz und einer Verzerrung der Kovarianzen, wodurch die ursprüngliche Datenverteilung nicht mehr adäquat abgebildet wird (vgl. Abschnitt 3.3.3). Die Stärke der Mittelwertimputation liegt hingegen vor allem in der hohen Rechenökonomie und Verfügbarkeit in `scikit-learn`, weshalb sie als geeignetes Verfahren im Experimentierplan berücksichtigt wird.

Die Medianimputation weist gegenüber der Mittelwertimputation Vorteile bei schiefen Verteilungen und Ausreißern auf, da der Median als robuste Lagekennzahl weniger empfindlich reagiert (vgl. Abschnitt 3.3.3). Dennoch bleibt auch hier eine Verzerrung der Datenstruktur bestehen, weil auch hier je Merkmal stets derselbe Wert imputiert wird. Aufgrund der ebenfalls geringen Rechenlast, hohen Interpretierbarkeit und Verfügbarkeit in `scikit-learn` ist den Kriterien nach geeignet für die Berücksichtigung im Experimentierplan.

Die Konstantenimputation wird deutlich kritischer bewertet. Zwar ist sie technisch einfach, verfügbar in `scikit-learn` und extrem recheneffizient, jedoch sind die erzeugten Ersatzwerte arbiträr und können die Datenstruktur stark verfälschen (vgl. Abschnitt 3.3.3). Ihre Berücksichtigung im Experimentierplan liegt primär darin, eine Vergleichsbasis für triviale Imputationsverfahren zu schaffen.

Die kNN-Imputation erzielt grundsätzlich gute Ergebnisse bei strukturierten numerischen Daten, da sie lokale Ähnlichkeiten berücksichtigt (vgl. Abschnitt 3.3.6). Hinsichtlich der Robustheit ist das Verfahren jedoch nur teilweise geeignet, da es empfindlich auf Ausreißer reagiert und in hochdimensionalen Datensätzen durch den Verlust an Trennschärfe des Distanzmaßes an Aussagekraft verliert (vgl. Abschnitt 3.3.6). Auch die Rechenökonomie ist lediglich teilweise erfüllt, da mit zunehmender Datenmenge für jede Imputation zahlreiche Distanzberechnungen erforderlich werden und der Aufwand dadurch erheblich steigt. Schließlich ist auch die Interpretierbarkeit eingeschränkt, da die Schätzwerte ausschließlich aus lokalen Nachbarschaftsbeziehungen abgeleitet werden und keine expliziten Modellparameter oder nachvollziehbaren Entscheidungsregeln vorliegen (vgl. Abschnitt 3.3.6). Trotz dieser Einschränkungen ist die kNN-Imputation für den vorliegenden Experimentierplan gut geeignet, da sie unkompliziert implementierbar ist und durch die Nutzung lokaler Strukturen einen praxisnahen und robusten Kontrast zu den einfacheren statistischen Verfahren darstellt.

Die iterative Imputation wird insgesamt positiv bewertet, da sie multivariate Zusammenhänge berücksichtigt und dadurch eine realitätsnahe Rekonstruktion fehlender Werte ermöglicht (vgl. Abschnitt 3.3.5). Einschränkungen bestehen jedoch hinsichtlich der Robustheit, da die Ergebnisse stark von Modellannahmen sowie der Konvergenz des iterativen Prozesses abhängen und bei kleineren Datensätzen instabil sein können. Die Rechenökonomie ist gering, da die mehrfachen Modellschätzungen mit einem erheblich höheren Rechenaufwand verbunden sind und das Verfahren dadurch deutlich weniger effizient als statistische Verfahren ist (vgl. Abschnitt 3.3.5). Zudem ist die Interpretierbarkeit eingeschränkt, da die einzelnen Regressionsschritte und Abhängigkeiten zwischen Variablen nicht transparent nachvollzogen werden können. Trotz dieser Limitierungen erweist sich die iterative Imputation als geeignetes Verfahren für den Experimentierplan, da sie multivariate Strukturen nutzt, in `scikit-learn` verfügbar ist und eine methodisch fundierte Alternative zu den einfacheren Verfahren bietet.

6.3.2 Auswahl der Verfahren zur Datenreduktion

Nach der Auswahl geeigneter Verfahren zur Behandlung fehlender Werte werden im nächsten Schritt Verfahren zur Datenreduktion ausgewählt, die in der Implementierung des Experimentierplans berücksichtigt werden. Die Auswahl berücksichtigt sowohl Verfahren zur Merkmals- als auch zur Instanzenreduktion (vgl. Abschnitt 4.1), da beide Ansätze auf die Reduktion redundanter Daten abzielen. Entscheidend für die Auswahl sind auch hier die in Abschnitt 5.2 definierten Kriterien zur Auswahl geeigneter DVV.

Als grundlegendes Filterverfahren wird die CFS einbezogen, welche Merkmale anhand ihres Zusammenhangs mit der Zielvariable und ihrer wechselseitigen Unabhängigkeit auswählt (vgl. Abschnitt 4.2.1). Obwohl keine native Implementierung in `scikit-learn` vorliegt, kann CFS durch eine Kombination aus Korrelationsanalyse und heuristischer Merkmalbewertung effizient nachgebildet werden. Es ist modular einsetzbar und geeignet für hochdimensionale Datensätze, bei denen Redundanzstrukturen häufig auftreten. Die hohe Interpretierbarkeit und geringen Rechenanforderungen durch die Verwendung statistischer Maße machen CFS zu einem praxistauglichen Verfahren für die systematische Merkmalsreduktion.

Außerdem wird MI eingesetzt, welches als informationsbasiertes Verfahren auch nichtlineare Zusammenhänge zwischen Merkmalen erkennen kann (vgl. Abschnitt 4.2.2). Die Umsetzung über `sklearn.feature_selection.mutual_info_classif` aus `scikit-learn` ermöglicht eine direkte Anwendung im Klassifikationskontext. MI überzeugt vor allem durch seine Robustheit gegenüber Datenvariationen sowie durch die Eignung für komplexe Merkmalsstrukturen und adressiert damit das Auswahlkriterium der Eignung für reale, nichtlinear strukturierte Daten.

Im Rahmen der Merkmalsreduktion wird auch der VIF eingesetzt, ein etabliertes Verfahren zur Erkennung linearer Redundanzen zwischen Prädiktoren (vgl. Abschnitt 4.2.3). Der VIF quantifiziert, in welchem Ausmaß ein Merkmal durch die übrigen erklärbar ist, und dient somit der gezielten Identifikation von Merkmalen, die keine eigenständige Information zum Modell beitragen. Die Umsetzung kann über eine eigene Implementierung unter Rückgriff auf die `statsmodels`-Bibliothek (statsmodels developers, 2023) erfolgen, da keine direkte Integration in `scikit-learn` besteht. Der VIF eignet sich insbesondere durch seine methodische Klarheit und hohe Interpretierbarkeit und wird damit als Verfahren zur Entfernung redundanter Merkmale im Experimentierplan berücksichtigt.

Die PCA (vgl. Abschnitt 4.2.6) erfüllt ebenfalls grundlegende Auswahlkriterien und wird im Experimentierplan berücksichtigt. Sie ermöglicht eine kompakte Repräsentation des Merkmalsraums unter Erhalt maximaler Varianzanteile und ist in `sklearn.decomposition.PCA` effizient implementiert. Obwohl die erzeugten Hauptkomponenten nicht mehr direkt interpretierbar sind, bietet PCA eine strukturierte und verlustarme Verdichtung des Merkmalsraums. Das Verfahren ist effizient implementierbar, rechenökonomisch und besonders geeignet für Datensätze mit hoher Multikollinearität.

Als Verfahren zur Reduktion von Instanzen wird schließlich ENN (vgl. Abschnitt 4.3.3) berücksichtigt. Es ist in `imblearn.under_sampling.EditedNearestNeighbours` verfügbar und entfernt potenziell fehlerhafte oder widersprüchliche Instanzen anhand lokaler Nachbarschaftsverhältnisse. ENN bietet eine robuste Möglichkeit zur strukturerhaltenden Bereinigung von Trainingsdaten und lässt sich nahtlos mit SMOTE kombinieren. Die hohe Kompatibilität mit dem eingesetzten Klassifikationsmodell sowie die stabile Implementierung sprechen für die praktische Einsetzbarkeit im Experimentierplan. Distanzbasierte Verfahren werden nicht berücksichtigt, da ihre Aussagekraft in hochdimensionalen Datensätzen stark abnimmt (vgl. Abschnitt 4.3.1). Clustering-Verfahren werden nicht berücksichtigt, da sie Instanzen zu Repräsentanten aggregieren und damit die Datenstruktur verändern (vgl. Abschnitt 4.3.2). Dies erschwert die eindeutige Interpretation der Effekte und ist im Rahmen des Experimentierplans nicht zielführend.

Die Auswahl der fünf Verfahren deckt grundlegende methodische Perspektiven der Datenreduktion ab und erfüllt die in der Methodik definierten Auswahlkriterien in wesentlichen Punkten. Durch die Kombination von Filter-, und Wrapperverfahren, Merkmalsextraktion sowie Instanzenselektion wird eine methodische Vielfalt erreicht, die differenzierte experimentelle Analysen ermöglicht. Auch für diese Verfahren gibt die folgende Tabelle 6.2 einen Überblick und bewertet sie anhand der Auswahlkriterien.

Die Tabelle 6.2 veranschaulicht die Bewertung ausgewählter Verfahren hinsichtlich zentraler Kriterien, wobei volle, halbvolle und leere Kreise für eine gute, teilweise oder geringe Erfüllung stehen. CFS wird insgesamt positiv bewertet, zeigt jedoch in drei Kategorien Einschränkungen. Die Eignung für reale Daten ist begrenzt, da das Verfahren von bedingter Unabhängigkeit der Merkmale ausgeht und bei komplexen Interaktionen an seine Grenzen stößt. Die Robustheit ist eingeschränkt, da sich die Merkmalsauswahl bei geringen Datenänderungen verändern kann. Auch die Verfügbarkeit ist nur teilweise gegeben, da es keine etablierte Python-Implementierung in gängigen Bibliotheken gibt. Allerdings lässt sich CFS aufgrund seiner einfachen Struktur mit geringem Aufwand selbst programmieren, was eine teilweise Erfüllung rechtfertigt. In den übrigen Kriterien überzeugt CFS durch hohe Interpretierbarkeit, Rechenökonomie und gute Kombinierbarkeit.

MI wird in den meisten Kategorien positiv bewertet, weist jedoch bei Robustheit und Interpretierbarkeit Einschränkungen auf. Die Robustheit gegenüber Datenvariationen ist nur teilweise erfüllt, da die Schätzung der wechselseitigen Information sensibel auf Ausreißer und Stichprobengröße reagiert. Auch die Interpretierbarkeit ist eingeschränkt, da MI zwar quantitative Relevanzwerte für Merkmale liefert, jedoch ohne nachvollziehbare Schwellen oder strukturelle Auswahlregeln. Insgesamt erfüllt MI einen Großteil der Kriterien durch seine breite Anwendbarkeit, Modularität und Verfügbarkeit in etablierten Bibliotheken.

Tabelle 6.2: Bewertung der ausgewählten Datenreduktionsverfahren anhand methodischer Kriterien

Verfahren	Eignung für Realdaten	Robustheit	Modularität & Kombinierbarkeit	Rechenökonomie	Interpretierbarkeit	Verfügbarkeit
CFS	◑	◑	●	●	●	◑
MI	●	◑	●	●	◑	●
VIF	●	◑	●	◑	●	◑
PCA	●	◑	●	●	○	●
ENN	●	◑	●	●	◑	●

Das VIF-Verfahren weist in drei Kategorien Einschränkungen bei der Erfüllung der Kriterien auf. Die Robustheit ist begrenzt, da das Verfahren empfindlich auf Rauschen und starke Korrelationen bei kleinen Stichproben reagiert. Die Rechenökonomie ist ebenfalls nur teilweise gegeben, da für jedes Merkmal ein Regressionsmodell gegen alle übrigen Merkmale berechnet wird. Auch die Verfügbarkeit ist eingeschränkt, denn VIF ist zwar technisch verfügbar (z. B. über `statsmodels`), aber nicht so unkompliziert nutzbar wie andere Feature-Selection-Verfahren, die direkt in gängigen Machine-Learning-Workflows wie `scikit-learn` integriert sind. Insgesamt überzeugt das VIF-Verfahren insbesondere durch seine praxisnahe Anwendbarkeit und die weitgehend nachvollziehbare Erkennung redundanter Merkmale.

Die PCA wird in der Kategorie Interpretierbarkeit mit einer geringen Erfüllung bewertet, da die erzeugten Hauptkomponenten komplexe Linearkombinationen aller ursprünglichen Merkmale darstellen und somit einer direkten inhaltlichen Deutung weitgehend entzogen sind. Die Robustheit ist als teilweise erfüllt bewertet, da die PCA empfindlich auf Ausreißer, Rauschen und unskalierte Merkmale reagiert. Trotz dieser Einschränkungen ist PCA aufgrund ihrer rechnerischen Effizienz, weitreichenden Verfügbarkeit und ihres Potenzials zur Aufdeckung verborgener Strukturen ein geeignetes Verfahren der Merkmalsextraktion.

Abschließend wird das ENN-Verfahren überwiegend positiv bewertet, zeigt jedoch in der Kategorie Robustheit Einschränkungen. Zwar entfernt das Verfahren inkonsistente Instanzen und kann dadurch die Klassifikationsleistung verbessern, jedoch besteht die Gefahr, dass auch informative Instanzen entfernt werden, was insbesondere bei komplexen Entscheidungsgrenzen problematisch sein kann. Die Interpretierbarkeit ist ebenfalls nur teilweise gegeben, da die Entscheidungen auf lokaler Nachbarschaft basieren und nicht durch explizite Regeln oder modellhafte Strukturen nachvollziehbar dokumentiert werden. In den übrigen Kriterien überzeugt

ENN durch seine einfache Integration in Pipelines, geringe Rechenanforderungen und breite Verfügbarkeit in etablierten Bibliotheken.

Nach der konzeptuellen Bewertung der eingesetzten Verfahren folgt nun die praktische Umsetzung im Rahmen des Experimentierplans. Das folgende Kapitel beschreibt die technische Durchführung der Experimente sowie die konkrete Implementierung der Verarbeitungsschritte in Python.

6.4 Durchführung der Experimente

Die technische Umsetzung des Experimentierplans erfolgt in Python auf Basis der zuvor definierten Verfahren und Konfigurationen. Für jede Konstellation wird eine Vorverarbeitungspipeline umgesetzt, die je nach Versuch Imputation, Datenreduktion und die erforderliche SMOTE kombiniert. Die Implementierung ist modular aufgebaut und verwendet ausschließlich etablierte Bibliotheken (`scikit-learn`, `imblearn`, `NumPy`, `pandas`) und ist so gestaltet, dass Verfahren austauschbar und erweiterbar sind. Alle Imputationsverfahren werden deklarativ in Dictionaries registriert und anschließend in Einzelversuchen aufgerufen bzw. zu Versuchskombinationen zusammengesetzt. Dadurch lassen sich technisch zusätzliche Verfahren oder abgewandte Parameter mit minimalem Änderungsaufwand in den Experimentierplan integrieren. Um Reproduzierbarkeit und methodische Vergleichbarkeit sicherzustellen, werden alle zufallsbasierten Verfahren mit einem festen Startwert über `random_state` ausgeführt. Dies betrifft insbesondere SMOTE, den Klassifikator, die PCA sowie die iterative Imputation. Auf diese Weise wird gewährleistet, dass die Ergebnisse nicht durch unterschiedliche Zufallswerte verzerrt werden und die Unterschiede zwischen den Verfahren auf die Datenvorverarbeitung zurückzuführen sind. Darüber hinaus wurden die Verfahren jeweils mit Parametern konfiguriert, die einen begründeten Ausgleich zwischen Genauigkeit und Effizienz schaffen. So wurden bei der iterativen Imputation die Zahl der Iterationen auf drei begrenzt, um den Rechenaufwand im hochdimensionalen SECOM-Datensatz handhabbar zu halten und dennoch eine robuste Schätzung multivariater Zusammenhänge zu ermöglichen. Bei der PCA wird ein Varianzschwellenwert von 95 % gewählt, sodass einerseits die wesentlichen Strukturen der Daten erhalten bleiben und andererseits eine deutliche Reduktion der Dimensionalität erreicht wird. Die selbst implementierten Reduktionsverfahren wurden ebenfalls mit klar definierten Parametern ausgestattet. Das CFS-Verfahren verwendet eine feste Korrelationseinschränkung von 0,7, um Merkmale mit hoher gegenseitiger Abhängigkeit auszuschließen und gleichzeitig eine hinreichende Variabilität der ausgewählten Merkmale zu sichern. Der Schwellenwert liegt im Bereich mittlerer Zusammenhänge 4.2.1 und stellt damit einen Kompromiss dar, der Redundanzen reduziert, ohne die Merkmalsbasis unnötig stark einzuschränken. Das MI-Verfahren basiert auf der Auswahl der Merkmale mit der höchsten wechselseitigen Information zur Zielvariablen (vgl. Abschnitt 4.2.2). Um einerseits die Komplexität im hochdimensionalen SECOM-Datensatz zu reduzieren und andererseits genügend Merkmale für eine aussagekräftige Analyse zu bewahren, wird die Anzahl der ausgewählten Merkmale auf 50 begrenzt. Für das VIF-Verfahren wurde ein Schwellenwert von 5 gewählt, da Werte oberhalb dieses Grenzwertes als Hinweis auf problematische Multikollinearität gewertet werden (vgl. Abschnitt 4.2.3).

Zunächst werden die Rohdaten des SECOM-Datensatzes eingelesen und in Merkmale sowie Zielvariable unterteilt. Anschließend werden je nach Versuch die in Abschnitt 6.3 ausgewählten DVV angewendet. Dabei wird zwischen Einzelverfahren und Kombinationen unterschieden. Bei den Kombinationen wird jede Konstellation in beiden möglichen Reihenfolgen (DVV1 → DVV2 und DVV2 → DVV1) ausgeführt, um explizit mögliche Reihenfolgeeffekte zu analysieren (vgl. Abschnitt 5.3.3).

Trotz ermittelter Eignung ist eine Herausforderung des SECOM-Datensatzes die stark unausgeglichene Zielvariable, da fehlerhafte Bauteile nur in etwa 6,6 % der Instanzen vorkommen. Um dieses Ungleichgewicht zu korrigieren, wird in allen Konfigurationen das Verfahren SMOTE eingesetzt, da ansonsten die Gefahr bestünde, dass das Modell fast ausschließlich die Mehrheitsklasse vorhersagt. Grundsätzlich wird SMOTE am Ende der jeweiligen DVV angewendet, also nach der Imputation und gegebenenfalls nach der Datenreduktion. Auf diese

Weise beruht die künstliche Datengenerierung auf einer sauberen und konsistenten Datenbasis, was die Präzision des Verfahrens erhöht und gleichzeitig die Rechenlast reduziert, da nur die tatsächlich verwendeten Merkmale berücksichtigt werden. Eine Ausnahme bildet das ENN-Verfahren, das gezielt Instanzen aus den Daten entfernt. Hier wird SMOTE zuerst angewendet, um die gleiche ausgeglichene Datenbasis wie bei den übrigen Verfahren herzustellen, und ENN entfernt anschließend redundante Instanzen. Dadurch bleibt die Vergleichbarkeit mit anderen Konfigurationen gewährleistet.

Da weder SMOTE noch einige Reduktionsverfahren mit fehlenden Werten umgehen können, wird zusätzlich eine temporäre Dummy-Imputation eingesetzt. Fehlende Werte werden dabei zunächst durch einen konstanten numerischen Platzhalter ersetzt. Konkret wurde der Wert 999999 gewählt, da er außerhalb des Wertebereichs der ursprünglichen Messdaten liegt und keine Überschneidungen mit realen Ausprägungen auftreten. Auf diese Weise können alle Verfahren, die keine fehlenden Werte akzeptieren, ohne technische Fehler ausgeführt werden. Zwar führt die Verwendung von Platzhaltern dazu, dass SMOTE diese Werte formal in die Interpolation einbezieht (vgl. Abschnitt 2.3), wodurch die erzeugten Instanzen nicht vollständig den realen Daten entsprechen. Dennoch wurde die Dummy-Imputation bewusst gewählt, da sie eine technisch einfache und methodisch konsistente Möglichkeit darstellt, alle Verfahrenskombinationen im Experimentierplan vergleichbar und reproduzierbar auszuführen. Damit handelt es sich nicht um eine perfekte, wohl aber um eine pragmatische und im gegebenen Kontext vertretbare Lösung.

Die Evaluation der Modellgüte erfolgt mittels dreifacher Kreuzvalidierung (vgl. Abschnitte 2.2.2, 6.2). Der Datensatz wird nach der Anwendung von SMOTE in drei gleich große Folds unterteilt, sodass jede Teilmenge die durch SMOTE hergestellte ausgeglichene Klassenverteilung beibehält. In jedem Durchlauf werden zwei Folds zum Training und ein Fold zum Testen verwendet. Auf diese Weise wird sichergestellt, dass beide Klassen in allen Teilmengen gleichermaßen vertreten sind und die dreifache Wiederholung zufällige Verzerrungen in der Ergebnisbewertung reduziert. Die Wahl von drei Folds stellt einen methodischen Kompromiss dar. Mit einer höheren Anzahl an Folds ließe sich die Schätzung der Modellgüte zwar noch stabiler absichern, allerdings steigen damit sowohl die Rechenzeit als auch die Komplexität der Auswertung erheblich an (vgl. Abschnitt 2.2.2). Eine geringere Anzahl von Folds würde hingegen das Risiko erhöhen, dass die Ergebnisse stark von einer einzelnen zufälligen Aufteilung abhängen. Drei Folds bilden daher einen ausgewogenen Mittelweg, sie gewährleisten eine hinreichend robuste Schätzung der Modellgüte, ohne den Rechenaufwand übermäßig zu erhöhen, und sind damit insbesondere für hochdimensionale Datensätze wie SECOM praktikabel. Als Klassifikator wird durchgängig der `HistGradientBoostingClassifier` verwendet. Dieses Modell wurde bewusst gewählt, da es robust gegenüber hochdimensionalen numerischen Daten ist, keine Vorannahmen über die Verteilung der Variablen erfordert und fehlende Werte verarbeiten kann (vgl. Abschnitt 2.2.1). Durch die Wahl eines einheitlichen Modells können Unterschiede in den Ergebnissen eindeutig auf die Datenvorverarbeitung zurückgeführt werden, ohne dass zusätzliche Modellunterschiede als Störfaktor auftreten. Die Ergebnisse werden einheitlich in einer Tabelle dokumentiert und am Ende des Programmdurchlaufs gespeichert. Neben den Mittelwerten und Standardabweichungen der Gütemetriken Accuracy, Precision, Recall und F1-Score wird auch die Rechenzeit wird für jeden Versuch vollständig erfasst, beginnend jeweils vor Anwendung des ersten DVV und endend nach Abschluss der Modellanwendung im Rahmen der Kreuzvalidierung. Damit fließen sowohl die Aufwände der Datenvorverarbeitung als auch die Modelllaufzeit in die Bewertung ein. Das Einbeziehen des Modells ist notwendig, da die gewählten DVV die Größe und Struktur der Datenbasis maßgeblich verändern und sich dies direkt auf die Dauer des Modelltrainings und der Vorhersage auswirkt. Auf diese Weise wird die Effizienz eines Verfahrens nicht isoliert, sondern in ihrem praktischen Gesamtkontext beurteilt, was eine realistische Einschätzung des Verhältnisses von Modellgüte und Rechenökonomie ermöglicht. Das Ablaufdiagramm des Quellcodes in Abbildung 6.5 verdeutlicht die durchgeführten Schritte vom Einlesen der Daten bis zur Ergebnisaggregation.

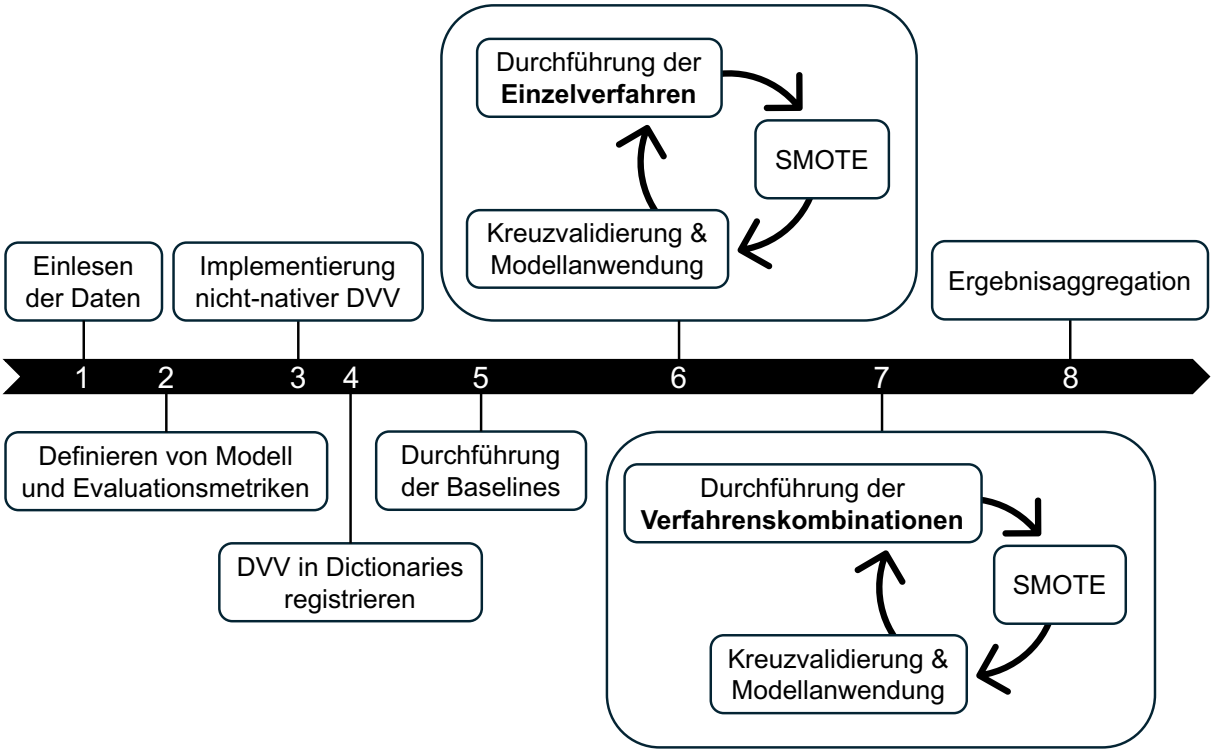


Abbildung 6.5: Ablaufdiagramm der experimentellen Durchführung

Abbildung 6.5 stellt den Ablauf der experimentellen Durchführung in acht Schritten anhand eines Ablaufdiagramms dar. Nach dem Einlesen der Rohdaten werden das Klassifikationsmodell und die Evaluationsmetriken definiert. Darauf folgt die Implementierung der DVV, die nicht direkt in Python-Bibliotheken verfügbar sind. Alle weiteren DVV werden wiederum in Dictionaries registriert, um später aufgerufen werden zu können. Anschließend erfolgt die Durchführung der Baselines, die als Referenz dienen. Im nächsten Schritt werden die Experimente mit den Einzelverfahren durchgeführt. Nach der Anwendung des jeweiligen DVV werden die Daten mit SMOTE ausbalanciert und anschließend im Rahmen einer dreifachen Kreuzvalidierung evaluiert. Anschließend werden die Verfahrenskombinationen getestet, wobei jede Kombination in beiden möglichen Reihenfolgen geprüft wird. Die in der Abbildung dargestellten Schleifen symbolisieren, dass alle Einzelverfahren und Verfahrenskombinationen jeweils systematisch nacheinander durchlaufen werden. Auch hier erfolgt nach der Datenvorverarbeitung die Anwendung von SMOTE und die anschließende Modellbewertung per Kreuzvalidierung. Abschließend werden die Ergebnisse aller Konfigurationen systematisch aggregiert und für die Auswertung bereitgestellt.

Nach der Darstellung des experimentellen Vorgehens und des Aufbaus der Programmlogik folgt nun die Auswertung der erzielten Resultate. Im folgenden Abschnitt werden die Ergebnisse der Experimente systematisch dargestellt und anhand der definierten Evaluationsmetriken beschrieben, um eine fundierte Grundlage für die anschließende Diskussion und Interpretation zu schaffen.

6.5 Auswertung der Versuchsergebnisse

In diesem Kapitel werden die im Rahmen des Experimentierplans erzielten Ergebnisse systematisch ausgewertet. Ziel der Analyse ist es, die Auswirkungen der Anwendung unterschiedlicher DVV auf die Leistungsfähigkeit des nachgelagerten Klassifikators zu untersuchen. Dabei wird sowohl die Modellgüte sowie die Stabilität der Ergebnisse als auch die benötigte Rechenzeit betrachtet, um neben der Vorhersagequalität auch Effizienzunterschiede zu bewerten. Für die Auswertung der Einzelverfahren dient die Baseline unter Einsatz von SMOTE als Referenz,

sodass die isolierten Effekte der verschiedenen Imputations- und Reduktionsverfahren im Vergleich zum Ausgangsmodell sichtbar werden. Bei den Kombinationsversuchen hingegen bilden die jeweiligen Einzelverfahren die Referenzpunkte. Auf diese Weise lassen sich die relativen Effekte der Verfahrenskombinationen und insbesondere potenzielle Reihenfolgeeinflüsse nachvollziehen. Die Auswertung erfolgt in zwei Schritten. Zunächst werden die Einzelverfahren betrachtet und im Anschluss die Kombinationen, sodass sowohl die isolierte Wirkung als auch die Wechselwirkungen der Verfahren auf deutlich werden.

6.5.1 Auswertung der Einzelverfahren

Zur Bewertung der einzelnen Imputationsverfahren werden die Ergebnisse jeweils mit der Baseline unter Einsatz von SMOTE verglichen, welches auch bei jedem anderen Versuch angewendet wurde. Abbildung 6.6 fokussiert sich auf die in Abschnitt 5.1.4 definierten primären Metriken und zeigt somit die erzielten Mittelwerte der Gütemaße Accuracy und F1-Score sowie die jeweils benötige Rechenzeit für Datenvorverarbeitung und Modellanwendung. Precision und Recall sowie die Standardabweichung der Gütemetriken wurden aus Gründen der Übersichtlichkeit nicht in das Diagramm aufgenommen, da ihre Aussage im F1-Score bereits in einer ausgewogenen Maßzahl zusammengeführt wird. Die detaillierten Ergebnisse sind jedoch in der Ergebnistabelle in Anhang B vollständig dokumentiert.

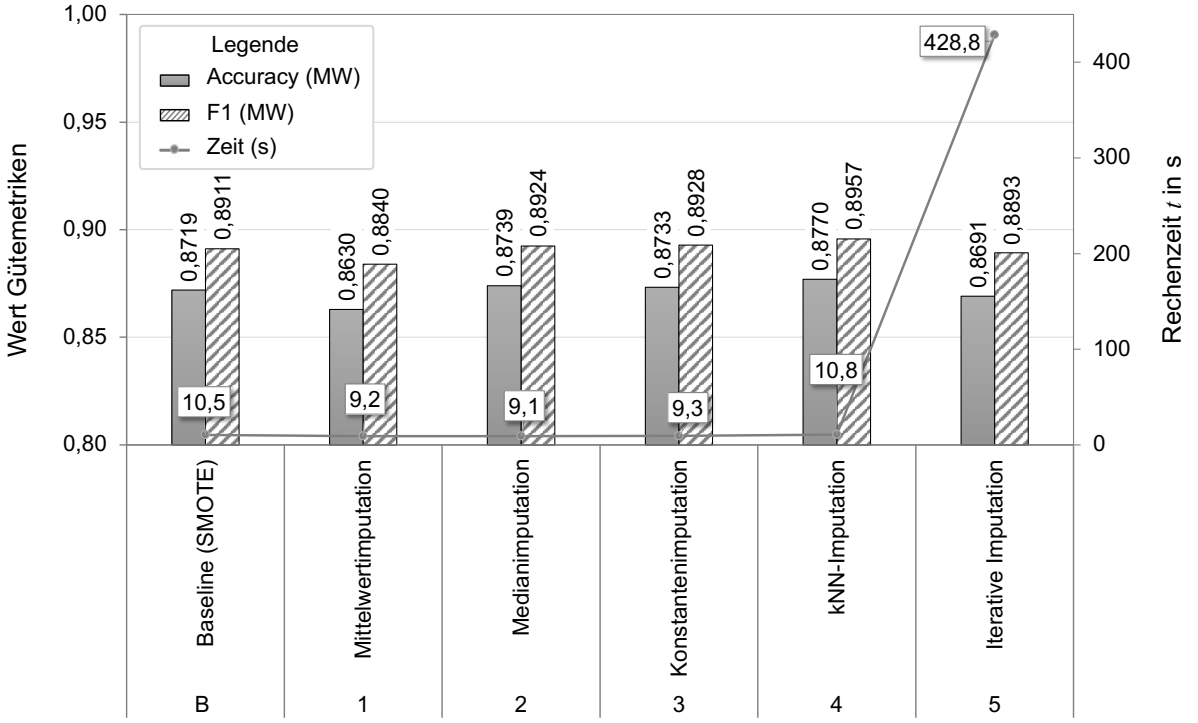


Abbildung 6.6: Versuchsergebnisse der einzeln angewandten Imputationsverfahren

Die Auswertung in Abbildung 6.6 vergleicht die fünf eingesetzten Imputationsverfahren mit der Baseline unter Verwendung von SMOTE. Auf der linken Achse sind Accuracy und F1-Score als Balken dargestellt, während die rechte Achse die Rechenzeit in Sekunden als Linie abbildet. Es zeigt sich in der Abbildung, dass die Accuracy-Werte aller Verfahren dicht beieinander liegen und sich im Bereich von etwa 0,86 bis 0,88 bewegen. Auch die F1-Scores weisen nur geringe Unterschiede auf und liegen zwischen 8,88 und 0,9. Deutlicher fallen hingegen die Unterschiede in der Rechenzeit aus. Während die einfacheren statistischen Verfahren sowie die kNN-Imputation mit Laufzeiten um 9 bis knapp 11 Sekunden nahe an der Baseline liegen, hebt sich die iterative Imputation mit einer Laufzeit von über 400 Sekunden deutlich ab. Zusammenfassend lässt sich festhalten, dass sich die statistischen Imputationsverfahren in Bezug auf die Modellgüte kaum von komplexeren Ansätzen, während die iterative Imputation zwar vergleichbare Werte liefert, jedoch mit einem erheblich höheren Rechenaufwand verbunden ist.

Im nächsten Schritt werden die Verfahren zur Datenreduktion betrachtet, um deren Einfluss auf Modellgüte und Effizienz zu analysieren. Abbildung 6.7 stellt dazu die Mittelwerte von Accuracy und F1-Score sowie die jeweilige Rechenzeit der Verfahren dar. Analog zur Auswertung der Imputationsverfahren liegt der Fokus im Diagramm auf den in Abschnitt 5.1.4 definierten primären Metriken, die vollständigen Versuchsergebnisse sind jedoch in der Ergebnistabelle in Anhang B dokumentiert.

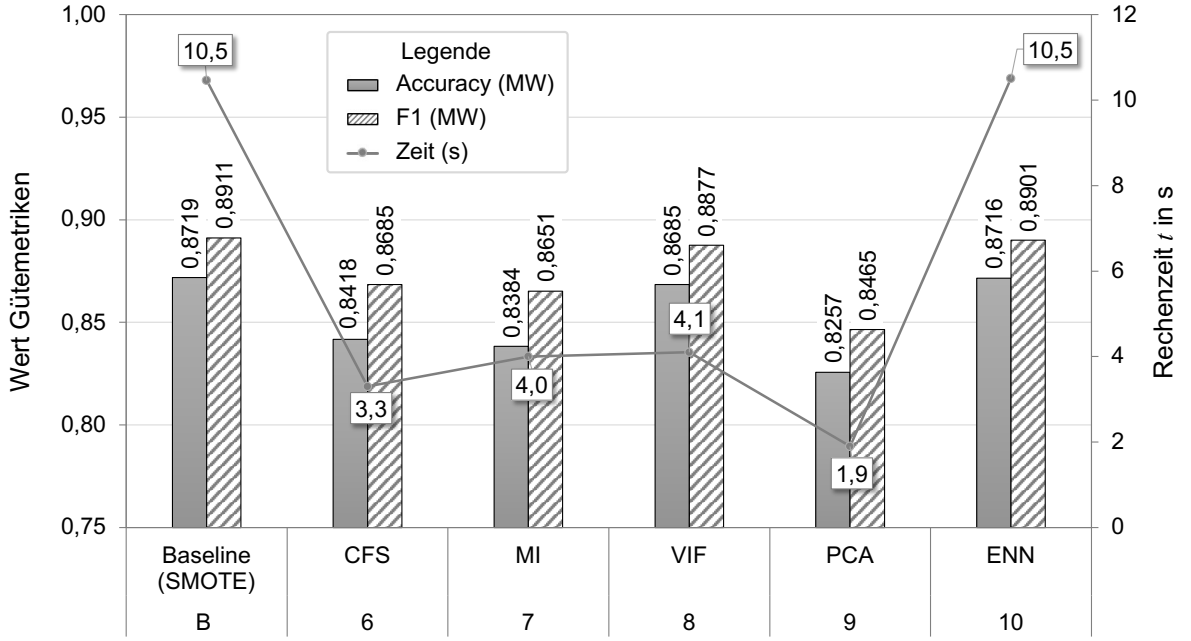


Abbildung 6.7: Versuchsergebnisse der einzeln angewandten Reduktionsverfahren

Die Auswertung in Abbildung 6.7 stellt die Ergebnisse der untersuchten Datenreduktionsverfahren im Vergleich zur Baseline mit SMOTE dar. Auf der linken Achse sind wieder Accuracy und F1-Score als Balken dargestellt, während die rechte Achse die benötigte Rechenzeit in Sekunden als Linie zeigt. Während VIF und ENN mit Accuracy-Werten von 0,8685 bzw. 0,8716 sowie F1-Scores von 0,8877 bzw. 0,8901 nahezu das Niveau der Baseline erreichen, fallen CFS und MI etwas ab. Besonders bei MI ist der Rückgang deutlich, mit Accuracy 0,8384 und F1-Score 0,8651. PCA erzielt mit Accuracy 0,8257 und F1-Score 0,8465 die schwächsten Ergebnisse und zeigt damit, dass die Verdichtung des Merkmalsraums zulasten der Modellleistung geht. Auffällig ist zudem, dass bei CFS und MI der Abstand zwischen Accuracy und F1-Score größer ausfällt, was auf eine höhere Sensitivität gegenüber unausgeglichenen Klassen hinweist. Auch bei der Laufzeit treten Unterschiede auf, die allerdings weniger ausgeprägt sind als bei den Imputationsverfahren. Mit 1,9 s weist PCA die kürzeste Rechenzeit auf, gefolgt von CFS mit 3,3 s, MI mit 4,0 s und VIF mit 4,1 s. ENN benötigt mit 10,5 s genauso viel Rechenzeit wie der Baseline-Versuch. Damit wird deutlich, dass vor allem PCA durch ihre kurze Laufzeit auf Kosten der Modellgüte hervorsticht, während VIF und ENN die besten Werte bezüglich der resultierenden Modellgüte erzielen.

Nachdem die Imputations- und Reduktionsverfahren zunächst isoliert betrachtet wurden, folgt im nächsten Abschnitt die Analyse ihrer Kombinationen im Experimentierplan. Ziel ist es, zu untersuchen, ob und inwiefern sich durch die Kombination von Imputations- und Reduktionsverfahren zusätzliche Effekte auf Modellgüte und Effizienz ergeben und inwieweit die Reihenfolge der Anwendung dabei eine Rolle spielt.

6.5.2 Auswertung der Verfahrenskombinationen

In diesem Abschnitt werden die Ergebnisse der Kombination von Imputations- und Reduktionsverfahren im Rahmen des Experimentierplans betrachtet. Dabei steht im Fokus, wie sich die aufeinander folgende Anwendung der Verfahren beider Ansätze auf die Effizienz und resultierende Modellgüte der nachgelagerten Klassifikators auswirkt. Durch die Analyse verschiedener Konstellationen wird ersichtlich, wie sich die Wirkung der Einzelverfahren verändert, wenn sie gemeinsam eingesetzt werden. Die Vorgehensweise bei der Auswertung erfolgt analog zur Auswertung der Einzelverfahren anhand der in Abschnitt 5.1.4 definierten primären Metriken. Dabei werden die Resultate gruppiert nach dem jeweils eingesetzten Imputationsverfahren dargestellt. Jede dieser Gruppen enthält dabei das Ergebnis des Einzelverfahrens sowie die der zugehörigen Kombinationen mit den Reduktionsverfahren in beiden geprüften Reihenfolgen. Auf diese Weise können Unterschiede innerhalb der Gruppen sowie potenzielle Reihenfolgeeffekte klar nachvollzogen werden. Abbildung 6.8 zeigt die Ergebnisse der Mittelwertimputation in Kombination mit den im Experimentierplan berücksichtigten Datenreduktionsverfahren.

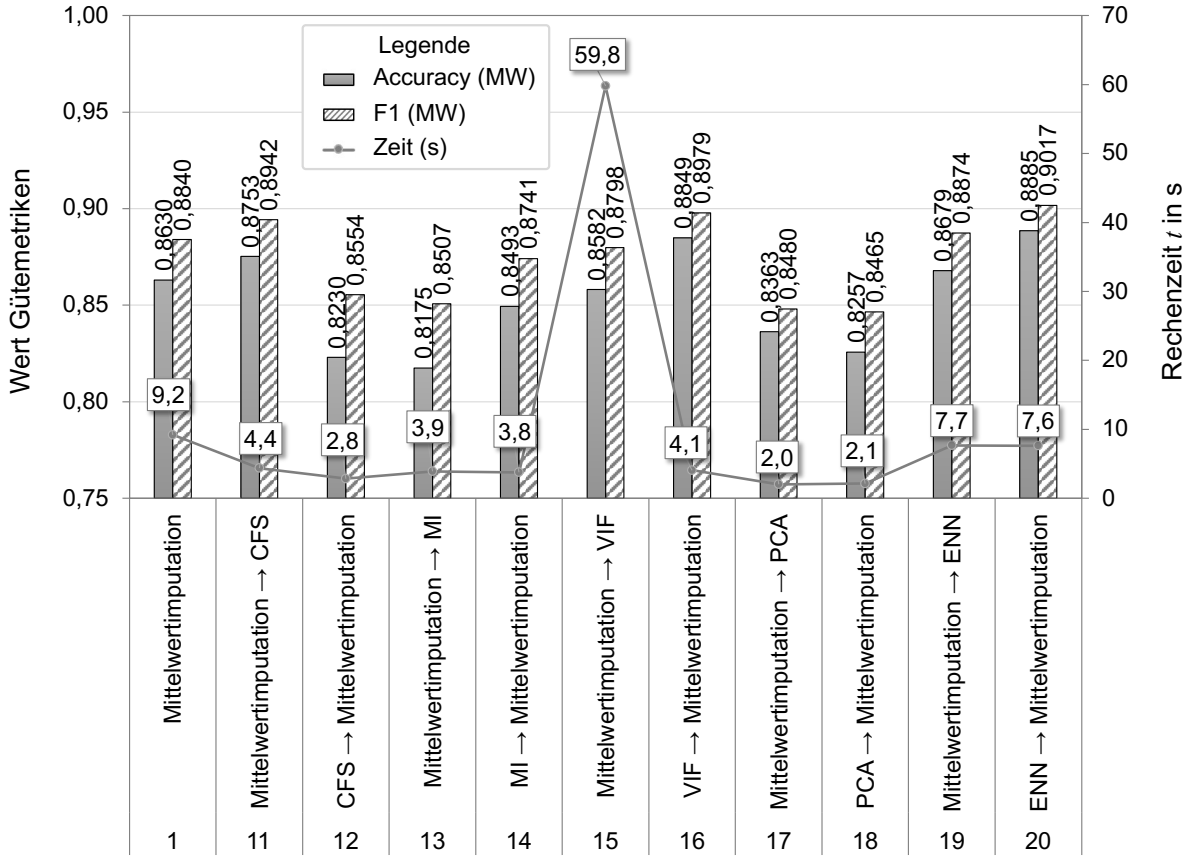


Abbildung 6.8: Versuchsergebnisse der Verfahrenskombinationen mit der Mittelwertimputation

Die in Abbildung 6.8 dargestellten Ergebnisse zeigen, dass die Kombination der Mittelwertimputation mit den Datenreduktionsverfahren zu teils deutlichen Unterschieden in der Modellgüte und Rechenzeit führt. Ein Beispiel hierfür ist die Kombination von Mittelwertimputation und CFS. Wird CFS im Anschluss an die Imputation durchgeführt, liegt der F1-Score bei 0,8942, während er bei umgekehrter Reihenfolge auf 0,8554 sinkt. Die Rechenzeit sinkt von 4,4 auf 2,8 Sekunden. Auch bei der Kombination mit MI zeigen sich klare Reihenfolgeeffekte. Hier erzielt Versuch 14, bei dem zuerst MI und anschließend die Mittelwertimputation durchgeführt wird, sichtbar bessere Ergebnisse hinsichtlich der Accuracy und F1-Score als der Gegenversuch 13. Die Rechenzeit hingegen ist in beiden Versuchen fast gleich. Ein besonders signifikanter Effekt ergibt sich bei der Kombination mit dem VIF. Wird der VIF vor der Imputation eingesetzt, erreicht der nachgelagerte Klassifikator nicht nur eine höhere Modellgüte

(F1: 0,8979 gegenüber 0,8798), sondern benötigt auch deutlich weniger Rechenzeit (4,1 s gegenüber 59,8 s). Die Kombination mit dem ENN-Verfahren liefert insgesamt die besten Ergebnisse in der betrachteten Gruppe. Insbesondere wenn ENN vor der Imputation angewendet wird, ergibt sich mit einem F1-Score von 0,9017 der höchste Wert aller getesteten Konfigurationen. Weniger ausgeprägt sind die Reihenfolgeeffekte bei der PCA. Beide Versuche liegen in ihrer Performance dicht beieinander, wobei beide unterhalb der Werte des Versuchs mit isolierter Mittelwertimputation (Versuch 1) liegen.

Auffällig ist zudem, dass die Precision-Werte im Vergleich zu Recall und Accuracy teils stärker schwanken (vgl. Anhang B). So variiert die Precision etwa zwischen 0,7481 (Versuch 13) und 0,8399 (Versuch 20), während der Recall in nahezu allen Versuchen konstant sehr hoch bleibt, teils über 0,98. Dieses Muster deutet darauf hin, dass die Modelle insgesamt viele tatsächliche Positivfälle korrekt erkennen (hoher Recall), dabei jedoch unterschiedlich gut darin sind, falsch-positive Vorhersagen zu vermeiden (schwankende Precision). In der praktischen Anwendung bezogen auf den SECOM-Datensatz kann dies bedeuten, dass ein Modell zwar nur wenige bestandene Instanzen verpasst, aber unter Umständen viele defekte Instanzen fälschlich als positiv klassifiziert, was je nach Einsatzgebiet problematisch sein kann. Auch die Standardabweichungen, insbesondere bei Precision und F1-Score, unterscheiden sich je nach Verfahren deutlich und geben Hinweise auf die Stabilität der Modelle über verschiedene Folds hinweg. Eine hohe Standardabweichung, wie etwa in Versuch 12 (CFS vor Mittelwertimputation) mit einer Precision-Standardabweichung von 0,1475, zeigt, dass die Modellleistung über die verschiedenen Validierungsläufe stark schwankt. In anderen Fällen, wie bspw. in Versuch 14 (MI vor Mittelwertimputation), ist die Streuung deutlich geringer, was auf konsistentere Ergebnisse hinweist. Zusammenfassend zeigt sich, dass die Kombination und Reihenfolge der Verfahren in dieser Gruppe messbare Auswirkungen auf die Modellgüte hat. Neben Accuracy und F1-Werten unterscheiden sich dabei auch die Rechenzeiten teils erheblich, wie insbesondere die Versuche mit dem VIF-Verfahren veranschaulicht.

Die nächste betrachtete Gruppe ist die der Iterativen Imputation in Kombination mit den verschiedenen Reduktionsverfahren. Die iterative Imputation ersetzt fehlende Werte durch schrittweise Annäherung mittels Regressionsmodellen (vgl. Abschnitt 3.3.5) und wird im Folgenden in Kombination mit CFS, MI, VIF, PCA und ENN untersucht. Auch in dieser Gruppe wird zwischen der Anwendung des Reduktionsverfahrens vor bzw. nach der Imputation unterschieden. Die Ergebnisse dieser Kombinationen werden gemeinsam mit dem Einzelverfahren dargestellt, um Unterschiede innerhalb der Gruppe und potenzielle Reihenfolgeeffekte identifizieren zu können. Abbildung 6.9 zeigt die Resultate für diese Gruppe in einem Diagramm analog zu den bereits betrachteten Auswertungen.

Die in Abbildung 6.9 dargestellten zeigen demnach die Modellgüte und Rechenzeit der Kombination der iterativen Imputation mit den im Experimentierplan berücksichtigten Datenreduktionsverfahren. Auch in dieser Gruppe lassen sich klare Unterschiede in den Versuchsergebnissen erkennen. So zeigt sich bei der Kombination mit CFS, dass die Anwendungsreihenfolge einen deutlichen Unterschied bewirkt. Wird CFS nach der iterativen Imputation durchgeführt (Versuch 51), liegt der F1-Score bei 0,8894. Bei umgekehrter Reihenfolge (Versuch 52) wird hingegen ein erkennbar niedrigerer Wert von 0,8533 erreicht, während gleichzeitig die Accuracy von 0,8688 auf 0,8192 sinkt. Die Rechenzeit unterscheidet sich dabei in beiden Versuchen erheblich. In Versuch 51 beträgt sie 437,2 s, während sie in Versuch 52 auf 6,6 s absinkt. Der Reihenfolgenwechsel bedeutet hier also einen großen Effizienzgewinn, jedoch auf Kosten der Modellgüte.

Ein ähnliches Muster zeigt sich bei der Kombination mit MI. Die Reihenfolge MI vor iterativer Imputation (Versuch 54) liefert jedoch leicht bessere Ergebnisse mit einem F1-Score von 0,8542 (Accuracy: 0,8240). Bei umgekehrter Anwendungsreihenfolge (Versuch 53) liegt der F1-Score bei 0,8480 (Accuracy: 0,8141), während die Rechenzeit auch hier in der Reihenfolge

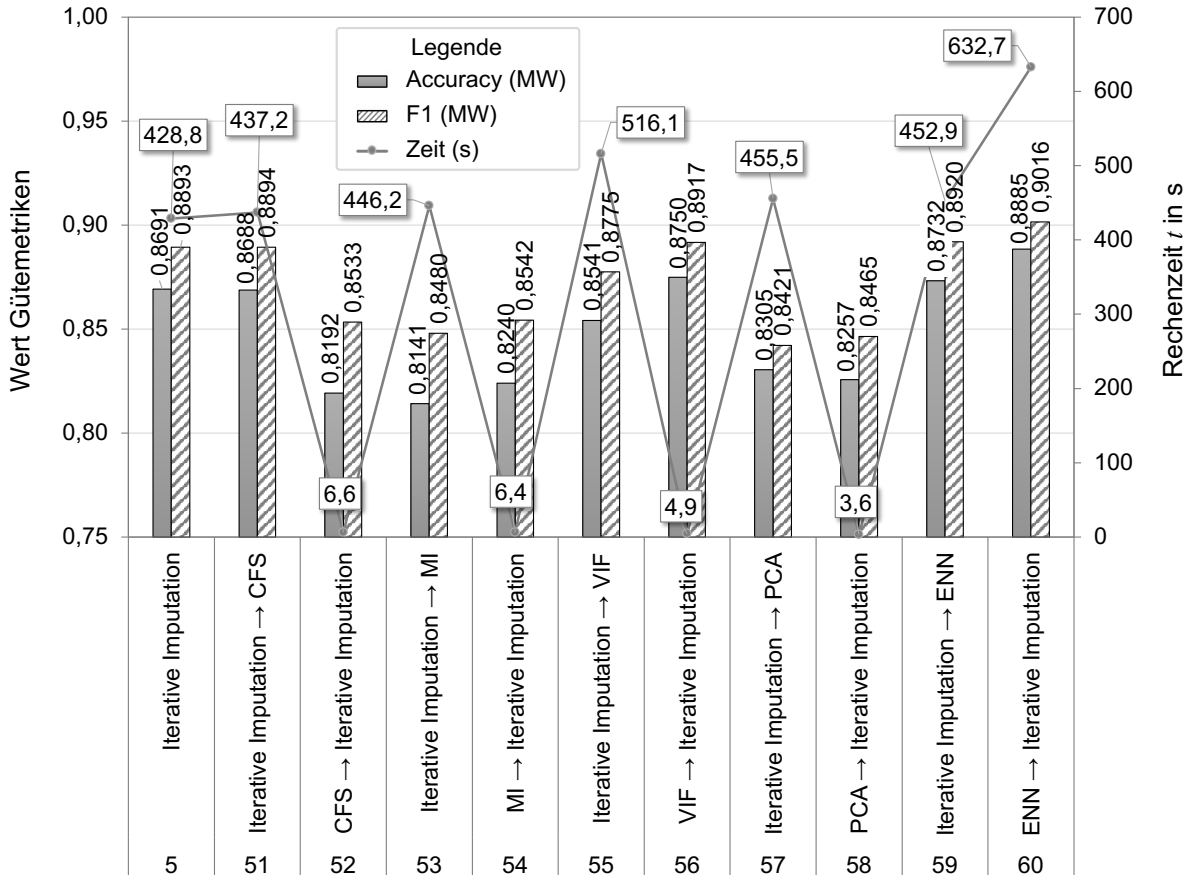


Abbildung 6.9: Versuchsergebnisse der Verfahrenskombinationen mit der iterativen Imputation

Imputation vor Reduktion mit 446,2 s im Vergleich zu 6,4 s deutlich höher liegt. Die Kombination mit dem VIF-Verfahren zeigt, wie bereits in der Gruppe der Mittelwertimputation, signifikante Unterschiede in Abhängigkeit von der Reihenfolge. Wird VIF vor der iterativen Imputation eingesetzt (Versuch 56), wird ein F1-Score von 0,8917 erreicht. Dieser liegt über dem Wert von 0,8775, der bei der umgekehrten Reihenfolge (Versuch 55) erzielt wird. Gleichzeitig sinkt die Rechenzeit von 516,1 s auf 4,9 s, was die Modellleistung Versuch 56 insgesamt sehr performant macht. Im Fall der PCA-Kombination liegen die Ergebnisse beider Versuche relativ nah beieinander. Versuch 57, bei dem die iterative Imputation vor der der PCA angewendet wird, erzielt einen F1-Score von 0,8421 (Accuracy: 0,8305). In Versuch 58 mit umgekehrter Reihenfolge liegt der Wert bei 0,8465 (Accuracy: 0,8257). Die Rechenzeiten unterscheiden sich jedoch auch hier mit 455,5 s gegen 3,6 s wieder stark. Die Ergebnisse der ENN-Kombination zeigen ein von den übrigen Kombinationen abweichendes Muster. Zugleich ist ENN das einzige Verfahren zur Reduktion von Instanzen im Experimentierplan. In beiden Versuchen ist die Modellgüte mit einer Accuracy um 0,88 und einem F1-Score um 0,9 vergleichsweise hoch. Jedoch ist hier auch die Rechenzeit mit 452,9 s (Versuch 59) und 632,7 s (Versuch 60) sehr hoch und fällt beim Reihenfolgenwechsel nicht ab.

Wie bereits in der zuvor betrachteten Gruppe zeigt sich auch bei dieser Gruppe, dass die Precision-Werte deutlich stärker schwanken als Recall und Accuracy (vgl. Anhang B). Die Precision reicht von 0,7487 in Versuch 53 bis 0,8386 in Versuch 60. Der Recall bleibt in nahezu allen Fällen konstant sehr hoch und liegt meist über 0,98. Dies deutet darauf hin, dass die Modelle zuverlässig positive Instanzen erkennen, aber unterschiedlich gut darin sind, falschpositive Vorhersagen zu vermeiden. Die Standardabweichungen, insbesondere bei Precision und F1-Score, unterscheiden sich ebenfalls deutlich und lassen Rückschlüsse auf die Modellstabilität zu. Beispielhaft sei Versuch 52 genannt, der mit einer Precision-Standardabweichung von 0,1463 relativ starke Schwankungen aufweist. Andere Konfigurationen, wie etwa Versuch 54, liefern deutlich stabilere Ergebnisse mit einer geringeren Precision-Standardabweichung

von 0,0953. Zusammenfassend wird in dieser Gruppe besonders deutlich, dass die Reihenfolge der Verfahren erhebliche Auswirkungen auf die Rechenzeit hat. In vielen Fällen führt eine geänderte Reihenfolge zu massiven Unterschieden im Zeitaufwand, teils bei vergleichbarer Modellgüte. Dies betrifft vor allem Kombinationen mit CFS, MI, VIF und PCA und zeigt die praktische Relevanz der Reihenfolgewahl bei rechenintensiven Verfahren wie der iterativen Imputation.

Im Anschluss wird die Gruppe der Konstantenimputation betrachtet. Dieses Imputationsverfahren ersetzt fehlende Werte durch einen festen, willkürlich gewählten Wert (hier 0) und wurde bewusst als triviales Beispiel in den Experimentierplan aufgenommen, um Unterschiede zu fundierteren Verfahren zu prüfen. Ziel ist es, zu untersuchen, wie sich eine derart vereinfachte Imputation in Kombination mit verschiedenen Reduktionsverfahren auf die Modellgüte und Rechenzeit auswirkt. Auch hier wird zwischen beiden Reihenfolgen der Anwendung unterschieden. Die Ergebnisse der Kombinationen werden zusammen mit dem Einzelverfahren dargestellt, um die Effekte innerhalb der Gruppe sichtbar zu machen. Eine Übersicht über die Resultate zeigt die folgende Abbildung 6.10.

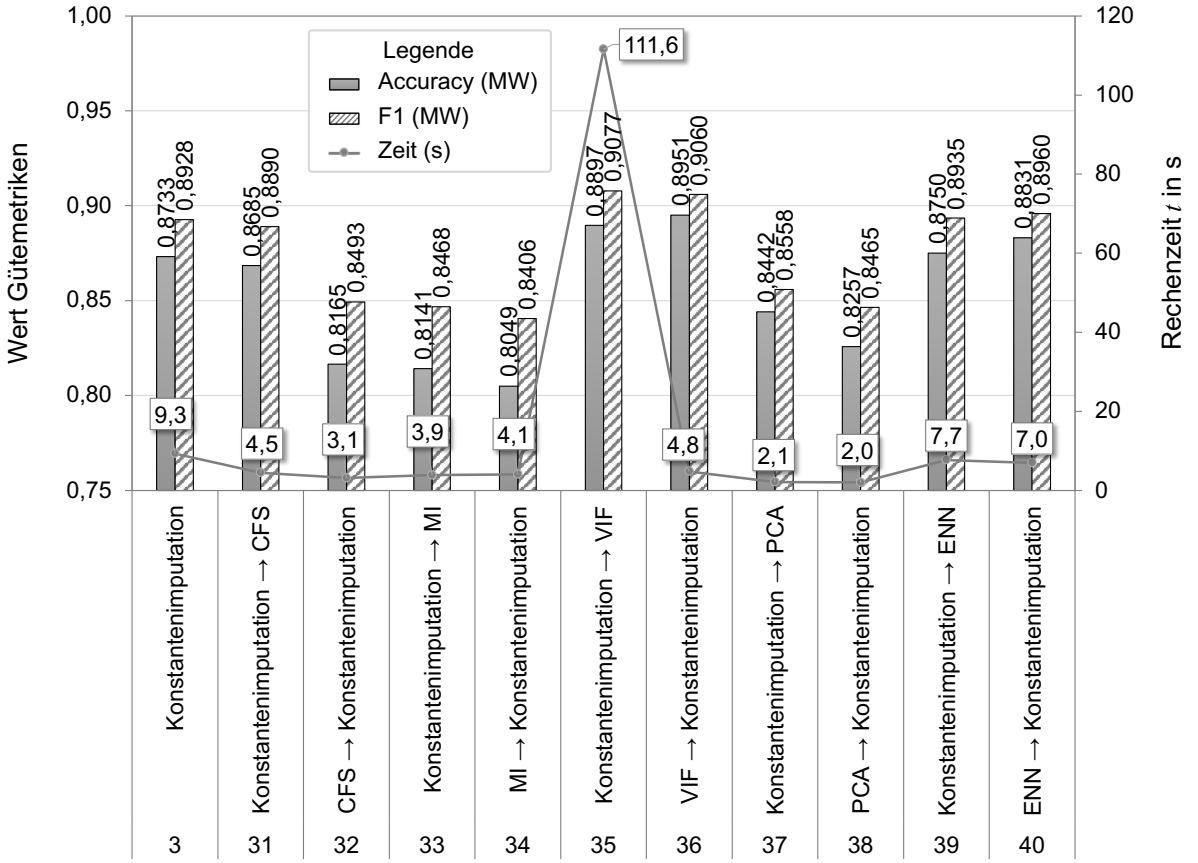


Abbildung 6.10: Versuchsergebnisse der Verfahrenskombinationen mit der Konstantenimputation

Abbildung 6.10 zeigt die Ergebnisse der Konstantenimputation in Kombination mit Reduktionsverfahren, jeweils in beiden getesteten Reihenfolgen. Insgesamt ähneln die Ergebnisse weitgehend denen der Gruppe mit Mittelwertimputation. Die F1-Werte bewegen sich in einem ähnlichen Bereich, wobei auch hier die besten Ergebnisse durch Kombination mit VIF und ENN erzielt werden. Versuch 35, in dem das VIF-Verfahren nach der Konstantenimputation angewendet wird, erreicht mit 0,9077 den höchsten F1-Score der gesamten Versuchsreihe. Der Gegenversuch 36 senkt mit einem fast gleichen Wert von 0,9060 die auffällig hohe Rechenzeit von 111,6 s auf 4,8 s. Bei den Kombinationen mit CFS, MI und PCA zeigt sich, dass die Variante mit Imputation vor Reduktion jeweils eine geringere Modellgüte erzielt als die umgekehrte Reihenfolge. Besonders drastisch fällt dieser Effekt in Kombination mit CFS auf (Versuch 31

und 32). Die Rechenzeiten innerhalb dieser Gruppe liegen insgesamt auf einem relativ niedrigen Niveau. Bei den Kombinationen mit CFS und MI bewegen sie sich um etwa 4 Sekunden, bei PCA sogar nur bei rund 2 Sekunden. In Kombination mit ENN betragen sie durchschnittlich etwa 7 Sekunden. Die Gruppe mit Konstantenimputation weist im Allgemeinen ein Ergebnisprofil auf, das dem der Mittelwertimputation sehr ähnlich ist. Auch hier treten in mehreren Fällen Unterschiede in der Modellgüte und Rechenzeit in Abhängigkeit von der Reihenfolge der Verfahren auf. Die Rechenzeiten bleiben dabei überwiegend niedrig, mit Ausnahme der Kombination mit VIF in Versuch 35. Bemerkenswert ist, dass trotz der methodischen Einfachheit des Verfahrens in mehreren Kombinationen Ergebnisse erzielt werden, die die Resultate komplexerer Imputationsverfahren erreichen oder sogar übertreffen.

Zur Vermeidung redundanter Darstellungen und übermäßiger Informationsdichte werden in diesem Abschnitt lediglich drei exemplarische Gruppen im Detail ausgewertet. Die Auswahl orientiert sich daran, dass sich anhand der Gruppen mit Mittelwert-, iterativer und Konstantenimputation bereits zentrale Muster und Effekte deutlich herausarbeiten lassen. Diese drei Konstellationen decken verschiedene methodische Ansätze ab und ermöglichen eine differenzierte Betrachtung typischer Reihenfolge- und Kombinationseffekte. Die vollständigen Ergebnisse der beiden verbleibenden Gruppen mit Median- und kNN-Imputation, einschließlich der jeweiligen Visualisierungen, sind im Anhang dokumentiert (vgl. Diagramme in Anhang C).

Die Analyse der Einzelverfahren und Verfahrenskombinationen im Experimentierplan zeigt klar, dass DVV und ihre Reihenfolge einen messbaren Einfluss auf die Modellleistung des nachgelagerten Klassifikators haben. In vielen Konstellationen ergeben sich deutliche Unterschiede bei der resultierenden Modellgüte und vor allem bei der Rechenzeit, was insbesondere bei rechenintensiven Verfahren wie der iterativen Imputation auffällt. Hier kann in vielen Fällen insbesondere die Anwendung der Reduktionsverfahren vor der Imputation den Zeitaufwand deutlich senken, ohne dass die Modellgüte nennenswert darunter leidet. Vergleichend wird deutlich, dass auch sehr einfache Verfahren wie die Konstantenimputation in Kombination mit effektiven Reduktionsverfahren zu konkurrenzfähigen Ergebnissen führen können. Gleichzeitig bestätigen sich bestimmte Verfahren wie VIF und ENN als besonders leistungsstark, unabhängig vom jeweiligen Imputationsverfahren. Die Erhebung zusätzlicher Metriken wie Standardabweichungen erweitert die Aussagekraft der Auswertung, da sie Rückschlüsse auf die Modellstabilität über verschiedene Folds der Kreuzvalidierung hinweg zulässt. Die drei exemplarisch ausgewerteten Gruppen mit Mittelwert-, iterativer und Konstantenimputation bilden eine ausgewogene Grundlage für die Identifikation typischer Muster und Effekte. Sie decken sowohl einfache als auch komplexe Imputationsverfahren ab und zeigen deutlich, wie stark die Kombination mit Reduktionsverfahren die Modellqualität beeinflussen kann. Um Redundanz zu vermeiden und die Lesbarkeit zu wahren, wurden die beiden verbleibenden Gruppen mit Median- und kNN-Imputation nicht im Haupttext analysiert. Ihre vollständigen Ergebnisse sowie die zugehörigen Diagramme befinden sich im Anhang und können dort nachvollzogen werden.

Die Auswertung der Versuchsergebnisse wird im folgenden Kapitel aufgegriffen und inhaltlich eingeordnet. Dabei werden die beobachteten Muster und Effekte vertiefend analysiert, ihre Ursachen diskutiert und bestehende Zusammenhänge herausgearbeitet. Darauf aufbauend erfolgt eine Bewertung der Ergebnisse im Hinblick auf ihre praktische Relevanz sowie eine Ableitung von Empfehlungen für den Einsatz geeigneter Verfahren und Reihenfolgen im Anwendungskontext.

7 Diskussion und Fazit

In diesem Kapitel werden die im Rahmen der Arbeit erzielten Ergebnisse systematisch aufgearbeitet und bewertet. Zunächst erfolgt eine Zusammenfassung der zentralen Resultate, bevor diese im Hinblick auf die Forschungsfrage interpretiert und ihre Ursachen sowie Implikationen diskutiert werden. Darauf aufbauend wird die methodische Vorgehensweise kritisch reflektiert, wobei sowohl Stärken als auch Limitationen der Arbeit herausgestellt werden. Den Abschluss bildet ein Fazit, das die wesentlichen Erkenntnisse bündelt und die erreichten Ziele der Untersuchung bewertet.

7.1 Zusammenfassung der Ergebnisse

Die durchgeführten Experimente zeigen, dass die untersuchten DVV die Ergebnisse von Data-Mining-Verfahren in unterschiedlichem Maße beeinflussen. Die Erwartung, dass komplexere Verfahren grundsätzlich zu präziseren Klassifikationen führen, konnte nicht bestätigt werden. Sowohl statistische Verfahren als auch anspruchsvollere Ansätze lieferten nahezu identische Gütemaße, der Unterschied lag hauptsächlich in der benötigten Rechenzeit. Während Mittelwert- und Median-, und Konstantenimputation mit sehr geringen Laufzeiten Ergebnisse auf hohem Niveau erzeugten, benötigte die iterative Imputation um ein Vielfaches mehr Rechenzeit. Dies verdeutlicht, dass die Komplexität eines Verfahrens nicht zwangsläufig einen Qualitätsgewinn mit sich bringt. Bei den Reduktionsverfahren zeigte sich, dass Verfahren wie VIF und ENN zu einer leichten Verbesserung der Modellgüte führten, dafür jedoch längere Laufzeiten in Kauf genommen werden mussten. Die PCA hingegen erzielte die kürzesten Rechenzeiten, blieb jedoch hinsichtlich der Klassifikationsergebnisse hinter den anderen Verfahren zurück. CFS und MI lagen in ihrer Leistung unterhalb der Baseline und konnten weder bei Effizienz noch bei Güte überzeugen.

Besonders auffällig waren die Ergebnisse bei den Verfahrenskombinationen und Reihenfolgevariationen. Wurde ein Reduktionsverfahren vor einem komplexeren Imputationsverfahren durchgeführt, konnte die Rechenzeit drastisch reduziert werden, ohne dass die Güte des Klassifikators darunter litt. Darüber hinaus zeigte sich, dass einfache Verfahren in Kombination mit geeigneten Reduktionsverfahren teilweise Ergebnisse erreichten, die mit den komplexeren Verfahren mithalten oder diese sogar übertreffen konnten. Zusätzlich offenbarte die Stabilitätsanalyse deutliche Unterschiede zwischen den getesteten Verfahren. Einige Konfigurationen lieferten konsistente Resultate über mehrere Durchläufe hinweg, während andere größere Schwankungen aufwiesen. Damit wurde neben Laufzeit und Modellgüte auch die Verlässlichkeit der Ergebnisse als relevante Dimension sichtbar. Insgesamt zeigt sich, dass die Wahl und Kombination der DVV maßgeblich die Resultate bestimmt und dass gerade im Zusammenspiel der Verfahren erhebliche Unterschiede sichtbar werden. Um diese Ergebnisse einordnen und ihre Ursachen besser verstehen zu können, ist eine vertiefte Interpretation erforderlich.

7.2 Interpretation der Ergebnisse

Die folgende Interpretation der Ergebnisse knüpft an die beobachteten Muster an und diskutiert, welche Eigenschaften der eingesetzten Verfahren die Ergebnisse erklären könnten und welche praktischen Implikationen sich daraus ableiten lassen. Zunächst werden die Einzelverfahren interpretiert, ehe die Analyse auf die Verfahrenskombinationen ausgeweitet wird.

Die Analyse der Imputationsverfahren macht deutlich, dass erhöhte Komplexität der DVV nicht automatisch mit einem Leistungsgewinn einhergeht. Die nahezu identischen Ergebnisse der Modellgüte von statistischen Verfahren und komplexeren Verfahren wie der iterativen Imputation lassen sich dadurch erklären, dass der zugrunde liegende Datensatz keine Strukturen enthält, die aufwändige Regressionsschätzungen erfordern. Fehlende Werte konnten offenbar schon durch unkompliziert berechnete Ersetzungen so ergänzt werden, dass der Klassifikator

die für die Zielvariable wesentlichen Muster zuverlässig erkennen konnte. Die iterative Imputation erzeugte daher keine messbare Verbesserung der Modellgüte, sondern lediglich einen extremen Anstieg der Rechenzeit. Für die praktische Anwendung lässt sich daraus ableiten, dass ein aufwändiges Verfahren nicht zwangsläufig bessere Ergebnisse liefert, sondern seine Stärke nur dann entfalten kann, wenn Daten hochgradig komplexe Abhängigkeiten enthalten, die einfache Schätzungen nicht abbilden können.

Bei den Reduktionsverfahren zeigte sich ein differenzierteres Bild. Dass VIF und ENN die Modellgüte leicht verbessern konnten, spricht dafür, dass diese Verfahren besonders effektiv darin sind, redundante oder störende Merkmale zu eliminieren und damit die Informationsdichte für den Klassifikator zu erhöhen. Dabei muss berücksichtigt werden, dass ENN lediglich redundante Instanzen entfernt und damit den Informationsgehalt hinsichtlich der Anzahl der Merkmale nicht mindert. Die PCA hingegen reduzierte zwar die Anzahl der Merkmale drastisch, entfernte dabei jedoch auch Informationen, die für die Klassifikation offenbar relevant gewesen wären. Das erklärt, warum PCA zwar die kürzeste Laufzeit aufwies, aber eine geringere Modellgüte erzielte. Die schwache Performance von CFS und MI hinsichtlich der resultierenden Modellgüte legt nahe, dass diese Verfahren für den SECOM-Datensatz praktisch nicht geeignet waren, weil ihre Kriterien für den Erhalt von Merkmalen nicht mit den tatsächlich relevanten Strukturen übereinstimmten. CFS beruht auf Korrelationen zwischen Merkmalen und dem Zielwert sowie zwischen den Merkmalen untereinander und erfasst damit vor allem lineare Abhängigkeiten. In einem hochdimensionalen Datensatz wie SECOM besteht jedoch die Gefahr, dass Merkmale ausgeschlossen werden, die zwar isoliert betrachtet keine starke Korrelation zeigen, aber in Kombination mit anderen Variablen wichtige Informationen enthalten. MI bewertet den Informationsgewinn jedes Merkmals einzeln, wodurch Variablen mit schwachem, aber komplementärem Beitrag verworfen werden, während redundante Merkmale mit hohem Einzelinformationswert erhalten bleiben. Auf diese Weise gingen dem Klassifikator entscheidende Signale verloren, was die Modellgüte verringerte. Die Ergebnisse machen deutlich, dass Verfahren mit univariaten oder ausschließlich korrelationsbasierten Auswahlkriterien bei hochkomplexen Datensätzen wie SECOM an ihre Grenzen stoßen und dass Ansätze besser geeignet sind, die auch Interaktionen und nichtlineare Zusammenhänge berücksichtigen können. Die Ergebnisse verdeutlichen, dass Datenreduktion nicht pauschal vorteilhaft ist, sondern je nach Verfahren und Datensatz sehr unterschiedliche Auswirkungen haben kann.

Noch deutlicher wurden die Effekte in der Verfahrenskombination mit Variation der Reihenfolge. Besonders die Beobachtung, dass VIF vor Imputation sowohl die Modellgüte steigerte als auch die Rechenzeit massiv senkte, verdeutlicht den praktischen Nutzen einer gezielten Reihenfolgegestaltung. Hier lässt sich die Ursache darin sehen, dass VIF zunächst die Zahl der Merkmale stark reduziert und damit die Imputation auf einen wesentlich kleineren Variablenraum angewendet wird. Dadurch sinkt die Rechenzeit erheblich, während die relevanten Informationen erhalten bleiben und die Güte sogar steigt. Ein ähnlicher Mechanismus erklärt die positiven Ergebnisse bei Kombinationen mit ENN, welches durch das Entfernen überflüssiger Instanzen die Klassifikation ohne Informationsverlust insgesamt stabiler und präziser macht. Besonders bemerkenswert ist, dass einfache Verfahren wie die Mittelwert- oder Konstantenimputation in Kombination mit Reduktionsverfahren Ergebnisse erreichten, die mit komplexeren Verfahren nicht nur mithalten, sondern sie teilweise übertreffen konnten. Das zeigt, dass die Stärke nicht in der isolierten Leistungsfähigkeit eines einzelnen Verfahrens liegt, sondern im Zusammenspiel innerhalb der Pipeline. Die Konstantenimputation in Kombination mit VIF, die überraschend den höchsten F1-Score der gesamten Untersuchung erzielte, verdeutlicht diesen Zusammenhang besonders stark. Obwohl die Ersetzung der fehlenden Werte durch einen festen Wert methodisch inadäquat erscheint, wurde ihre Wirkung durch die anschließende Anwendung von VIF deutlich abgeschwächt. VIF entfernte Merkmale, die durch die uniforme Ersetzung wenig informativ oder redundant geworden waren, sodass am Ende nur robuste und relevante Variablen für den Klassifikator verblieben. Dadurch verlor die wenig differenzierte Imputation an Einfluss und es konnten trotz der einfachen Vorgehensweise sehr gute Ergebnisse erzielt werden. Diese Konstellation zeigt, dass auch sehr einfache Verfahren, die methodisch wenig differenziert sind, in Kombination mit geeigneten Reduktionsverfahren

leistungsfähige Ergebnisse erzielen können. Die Konstantenimputation erreichte bereits isoliert betrachtet eine vergleichbare Modellgüte wie andere statistische Ansätze. In Verbindung mit VIF konnten die Ergebnisse jedoch noch einmal gesteigert werden, dass selbst triviale, vermeintlich inadäquate Verfahren in der richtigen Kombination eine hohe Leistungsfähigkeit entfalten können. Entscheidend ist daher nicht allein die Wirksamkeit einzelner Verfahren, sondern ihr gezieltes Zusammenspiel innerhalb der Pipeline.

Ein weiteres zentrales Ergebnis betrifft die Unterschiede zwischen Precision und Recall. Über nahezu alle Konfigurationen hinweg blieb der Recall konstant sehr hoch, was darauf hindeutet, dass der Klassifikator positive Fälle zuverlässig erkennt. Die teilweise stark schwankende Precision weist jedoch darauf hin, dass die Modelle unterschiedlich stark dazu neigen, falsch-positive Klassifikationen zu erzeugen. Dieses Muster lässt sich damit erklären, dass die Imputations- und Reduktionsverfahren die Entscheidungsgrenzen des Klassifikators unterschiedlich formen. Während manche Konfigurationen ein sehr robustes Abbild der relevanten Muster erzeugen, führen andere dazu, dass die Klassifikation unsauber trennt und damit vermehrt falsch-positive Ergebnisse liefert. Für die praktische Anwendung bedeutet das, dass ein hoher Recall allein nicht genügt, um ein Verfahren als gut zu bewerten, da eine hohe Rate an falsch-positiven Klassifikationen Prozesse ebenso stark belasten kann wie eine niedrige Trefferquote.

Schließlich liefert die Analyse der Standardabweichungen wichtige Hinweise auf die Stabilität der Ergebnisse der Data-Mining-Verfahren. Große Schwankungen, wie sie etwa bei der Kombination von CFS und Mittelwertimputation beobachtet wurden, deuten auf eine hohe Abhängigkeit von den einzelnen Folds der Kreuzvalidierung hin. Das bedeutet, dass die Modelle in Abhängigkeit von der konkreten Trainings- und Testaufteilung stark variieren. Während CFS stark auf Korrelationen zwischen Merkmalen reagiert und je nach Datenteilmenge unterschiedliche Merkmalsauswahlen trifft, könnte die Mittelwertimputation durch ihre Abhängigkeit von Ausreißern und die unterschiedliche Berechnung des Mittelwerts in den einzelnen Folds variierende Imputationswerte erzeugen. Dies lässt vermuten, dass sich beide Effekte gegenseitig verstärken und in den Ergebnissen als erhöhte Standardabweichungen der Gütemaße sichtbar werden. Umgekehrt zeigen Konfigurationen mit stabileren Ergebnissen, wie bspw. MI vor iterativer Imputation, dass Verfahren mit geringeren Schwankungen in der Praxis verlässlicher sind, selbst wenn ihre durchschnittliche Güte etwas geringer ist. Eine mögliche Erklärung hierfür ist, dass beide Verfahren auf plausiblen Schätzungen basieren, die im Gegensatz zu statistischen Imputationsverfahren weniger anfällig für Ausreißer oder Stichprobenvariationen sind. Für den praktischen Anwendungsfall ergibt sich daraus die Schlussfolgerung, dass Stabilität stets neben Modellgüte und Effizienz berücksichtigt werden muss. Ein Verfahren, das konsistente Ergebnisse liefert, ist oftmals geeigneter als eines, das in Einzelfällen etwas bessere Werte erreicht, dabei aber unvorhersehbar schwankt. Die zentralen Ergebnisse sind damit eingeordnet, woran sich eine kritische Reflexion der methodischen Vorgehensweise im folgenden Abschnitt anschließt.

7.3 Kritische Reflexion

Der Experimentierplan hat sich als geeignet erwiesen, um die Forschungsfrage nach den Effekten von Reihenfolge und Kombination von DVV systematisch zu beantworten. Die strukturierte und modulare Anlage ermöglichte es, Unterschiede in Effizienz und Modellgüte nachvollziehbar zu erfassen und zugleich Wechselwirkungen zwischen den Verfahren sichtbar zu machen. Insbesondere die Reihenfolgeeffekte ließen sich durch den Aufbau eindeutig erkennen, wodurch die Arbeit einen wichtigen Beitrag zur realistischen Bewertung von Datenvorverarbeitung leistet. Damit erfüllt die Arbeit nicht nur die inhaltlichen Zielsetzungen, sondern schließt auch eine Lücke im Stand der Technik, indem sie zeigt, wie sich durch die gezielte Variation von Verfahren innerhalb eines konsistenten und erweiterbaren Vergleichsrahmens praxisnahe Erkenntnisse über die Gestaltung von Data-Mining-Pipelines gewinnen lassen.

Gleichzeitig sind einige Einschränkungen zu berücksichtigen. Die Experimente wurden auf dem SECOM-Datensatz durchgeführt, der durch eine unausgeglichene Verteilung der Zielva-

riablen geprägt ist. Dieses Ungleichgewicht musste mit SMOTE künstlich ausgeglichen werden, wodurch die Klassifikationsaufgabe zwar handhabbar, aber zugleich durch künstlich erzeugte Daten beeinflusst wurde. Zudem war für die Anwendung der Reduktionsverfahren eine Dummy-Imputation erforderlich, da die meisten dieser Verfahren nicht mit fehlenden Werten umgehen können. Fehlende Werte wurden daher zunächst durch einen konstanten Platzhalter ersetzt. Da SMOTE diese Platzhalter jedoch in seine Interpolationen einbezieht, lassen sich die Werte nachträglich nicht in jedem Fall eindeutig wieder durch NaN kennzeichnen. Dies führt zu einer leichten Verzerrung der Datenbasis, die methodisch in Kauf genommen wurde, um die Vergleichbarkeit und Durchführbarkeit aller Kombinationen sicherzustellen. Es konnten außerdem nicht alle komplexeren Verfahren berücksichtigt werden, da deren Rechenzeit auf der verfügbaren Hardware ein praktikables Maß überschritten hätte. Stattdessen konzentrierte sich die Arbeit auf eine repräsentative Auswahl, die sowohl einfache als auch komplexere Ansätze abdeckt.

Die genannten Limitationen markieren die Rahmenbedingungen der Arbeit, beeinträchtigen jedoch nicht die Aussagekraft der zentralen Ergebnisse. Vielmehr verdeutlichen sie, unter welchen Rahmenbedingungen die Untersuchung durchgeführt wurde, und machen den spezifischen Erkenntnisgewinn transparent. Innerhalb dieses Rahmens liefert die Arbeit robuste und praxisnahe Ergebnisse, die klare Einsichten in die Wirkung von Imputations- und Reduktionsverfahren sowie deren Kombination und Reihenfolgevariation ermöglichen. Damit leistet die Arbeit einen Beitrag zur fundierten Bewertung von Strategien in der Datenvorverarbeitung und liefert praxisnahe Hinweise für den Einsatz datengetriebener Verfahren im industriellen Umfeld.

7.4 Fazit

Die Ergebnisse der Arbeit legen nahe, dass die Wahl und Reihenfolge von DVV einen messbaren Einfluss auf die Resultate nachgelagerter Klassifikationsverfahren haben können. Dieser Einfluss betrifft sowohl die Modellgüte als auch die Rechenzeit und die Robustheit der Ergebnisse. Während die Modellgüte bei vielen Verfahren ähnlich blieb, unterschieden sich die Verfahren deutlich in ihrer Effizienz und Verlässlichkeit. Damit wird deutlich, dass die Datenvorverarbeitung nicht nur ein vorbereitender Schritt ist, sondern eine zentrale Stellgröße, die die Gesamtleistung datengetriebener Analysen wesentlich beeinflussen kann.

Ein zentrales Ergebnis der Analyse ist der Hinweis darauf, dass komplexere DVV nicht zwangsläufig zu besseren Klassifikationsergebnissen führen. In vielen Fällen genügten bereits einfache statistische Verfahren zur Aufbereitung der Daten, um vergleichbare Resultate zu erzielen. Auffällig war zudem, dass einfache Verfahren in Kombination mit geeigneten Reduktionsverfahren teilweise sogar bessere Ergebnisse lieferten als deutlich komplexere modellbasierte Ansätze. Entscheidend ist also nicht die Komplexität eines einzelnen Verfahrens, sondern dessen Zusammenspiel mit anderen Verfahren innerhalb der Pipeline.

Die Analyse von Kombination und Reihenfolgenvariation deutet darauf hin, dass durch eine gezielte Anordnung der Verfahren erhebliche Effizienzgewinne möglich sind, ohne dass die Modellgüte darunter leidet. In einigen Konstellationen ließ sich die Güte sogar steigern, was den praktischen Nutzen einer bewussten Pipeline-Gestaltung unterstreicht. Datenvorverarbeitung sollte daher nicht als isolierte Wahl einzelner Verfahren verstanden werden, sondern als abgestimmtes Gesamtsystem, dessen Wirkung nur im Zusammenspiel beurteilt werden kann.

Die Arbeit leistet damit einen Beitrag, der über die Betrachtung isolierter Verfahren hinausgeht. Sie zeigt, dass systematische Kombinations- und Reihenfolgeeffekte einen wesentlichen Einfluss auf die Analyseergebnisse haben können und deren gezielte Betrachtung praxisnahe und differenzierte Einsichten ermöglicht. Damit konnten die Ziele der Arbeit erreicht werden, indem sowohl die Wirkung einzelner Verfahren als auch ihr Zusammenspiel untersucht und bewertet wurde. Auf dieser Grundlage folgt abschließend eine Zusammenfassung der wichtigsten Ergebnisse sowie ein Ausblick auf weiterführende Forschungsperspektiven.

8 Zusammenfassung und Ausblick

Die Arbeit hat untersucht, welchen Einfluss Verfahren der Datenvorverarbeitung auf die Ergebnisse von Klassifikationsverfahren haben. Betrachtet wurden dabei Verfahren zur Imputation fehlender Werte sowie zur Reduktion von Merkmalen und Instanzen, die einzeln, in Kombination und in variierenden Reihenfolgen analysiert wurden. Die Ergebnisse zeigen, dass die Wahl und Anordnung dieser Verfahren einen messbaren Einfluss auf Modellgüte, Rechenzeit und Stabilität hat. Während sich die Modellgüte bei vielen Verfahren auf einem ähnlichen Niveau bewegte, zeigten sich vor allem bei Laufzeit und Robustheit deutliche Unterschiede. Besonders deutlich wurde, dass komplexe modellbasierte Verfahren nicht zwangsläufig bessere Resultat erzielen, sondern einfache statistische Verfahren in vielen Fällen gleichwertige oder sogar bessere Voraussetzungen für präzise Klassifikationsergebnisse schaffen können. Von besonderer Bedeutung erwies sich die Reihenfolge der Verfahrensanwendung, da sie in mehreren Konstellationen Effizienzgewinne ermöglichten, ohne die Modellgüte zu mindern.

Die Ergebnisse der Arbeit haben sowohl theoretische als auch praktische Implikationen. Die Ergebnisse legen nahe, dass die Forschung zur Datenvorverarbeitung verstärkt das Zusammenspiel von Verfahren und deren Reihenfolge berücksichtigen sollte, da hier erhebliche Einflüsse auf Effizienz und Ergebnisqualität entstehen. Für die Praxis verdeutlichen die Ergebnisse, dass einfache, statistische Imputationsverfahren in Kombination mit gezielt gewählten Reduktionsverfahren oft eine solide und effiziente Lösung darstellen können. Die bewusste Gestaltung der gesamten Pipeline tritt damit stärker in den Vordergrund als die Wahl einzelner Verfahren. Die gewonnenen Erkenntnisse könnten dazu beitragen, Leitlinien für die Auswahl und Kombination von DVV zu entwickeln, die sich an unterschiedlichen Anwendungsszenarien orientieren. Die Ergebnisse können damit als Grundlage dienen, datengetriebene Prozesse systematischer, transparenter und effizienter zu gestalten.

Trotz dieser Erkenntnisse bleiben Fragen offen, die weiterer Forschung bedürfen. Die Untersuchung beschränkte sich auf einen Datensatz mit spezifischen Eigenschaften, dessen Klassenungleichgewicht durch künstliche Erweiterung ausgeglichen werden musste. Unklar ist, inwieweit die identifizierten Muster auch in Datensätzen mit abweichender Struktur, etwa aus anderen Domänen, mit anderer Verteilung oder größerem Datenvolumen, reproduzierbar sind. Zudem konnten nicht alle verfügbaren Verfahren einbezogen werden, insbesondere solche mit hohem Ressourcenbedarf. Die Experimente wurden ausschließlich mit einem Klassifikator durchgeführt, sodass nicht ausgeschlossen werden kann, dass andere Lernalgorithmen unterschiedlich stark auf Reihenfolge und Kombination der DVV reagieren.

Daraus ergeben sich mehrere Perspektiven für zukünftige Forschung. Eine naheliegende Richtung ist die Ausweitung auf unterschiedliche Datensätze, um die Generalisierbarkeit der Ergebnisse zu prüfen. Dabei erscheint es besonders relevant, nicht nur Daten mit ähnlicher Struktur, sondern auch andere Typen und Herkünfte einzubeziehen, etwa Logistikdaten oder Prozessdaten aus anderen industriellen Bereichen. Darüber hinaus könnte die Integration weiterer DVV, etwa solcher auf Deep-Learning-Ansätzen, das Spektrum der untersuchten Verfahren sinnvoll erweitern. Auch die systematische Analyse unterschiedlicher Klassifikatoren und deren Interaktion mit DVV eröffnet zusätzliche Erkenntnismöglichkeiten. Der entwickelte Quellcode ist bewusst modular aufgebaut, sodass sich weitere Verfahren mit geringem Anpassungsaufwand integrieren lassen. Damit kann die Arbeit als Grundlage für weiterführende Vergleiche dienen und eine flexible Umgebung zur Überprüfung zusätzlicher Ansätze bereitstellen.

Literaturverzeichnis

Aggarwal, C. C. (2018). An Introduction to Cluster Analysis. In C. C. Aggarwal & C. K. Reddy (Hrsg.), *Data Clustering: Algorithms and Applications* (S. 1–23). Chapman and Hall/CRC. https://doi.org/10.1201/9781315373515

Aggarwal, C. C., Hinneburg, A., & Keim, D. A. (2001). On the Surprising Behavior of Distance Metrics in High Dimensional Space. In J. Van den Bussche & V. Vianu (Hrsg.), *Database Theory—ICDT 2001* (S. 420–434). Springer Berlin Heidelberg.

Akinwande, M. O., Dikko, H. G., & Samson, A. (2015). Variance Inflation Factor: As a Condition for the Inclusion of Suppressor Variable(s) in Regression Analysis. *Open Journal of Statistics*, *05*(07), 754–767. https://doi.org/10.4236/ojs.2015.57075

Bankhofer, U. (1995). *Unvollständige Daten- und Distanzmatrizen in der multivariaten Datenanalyse* [Technische Universiät Ilmenau]. https://doi.org/10.22032/DBT.40346

Beaulieu-Jones, B. K., Lavage, D. R., Snyder, J. W., Moore, J. H., Pendergrass, S. A., & Bauer, C. R. (2018). Characterizing and Managing Missing Structured Data in Electronic Health Records: Data Analysis. *JMIR Medical Informatics*, *6*, e11. https://doi.org/10.2196/medinform.8960

Bekkar, M., Djemaa, H. K., & Alitouche, T. A. (2013). Evaluation Measures for Models Assessment over Imbalanced Data Sets. *Journal of Information Engineering and Applications*, *3*(10), 27–36.

Beraha, M., Metelli, A. M., Papini, M., Tirinzoni, A., & Restelli, M. (2019). Feature Selection via Mutual Information: New Theoretical Insights. *2019 International Joint Conference on Neural Networks (IJCNN)*, 1–9. https://doi.org/10.1109/IJCNN.2019.8852410

Bishop, C. M. (2006). *Pattern Recognition and Machine Learning*. Springer.

Blum, A. L., & Langley, P. (1997). Selection of Relevant Features and Examples in Machine Learning. *Artificial Intelligence*, *97*, 245–271. https://doi.org/10.1016/S0004-3702(97)00063-5

Bolón-Canedo, V., Sánchez-Maroño, N., & Alonso-Betanzos, A. (2013). A Review of Feature Selection Methods on Synthetic Data. *Knowledge and Information Systems*, *34*(3), 483–519. https://doi.org/10.1007/s10115-012-0487-8

Bordino, A., & Berrett, T. B. (2024). *Tests of Missing Completely At Random Based on Sample Covariance Matrices*. https://doi.org/10.48550/arXiv.2401.05256

Braiek, H. B., & Khomh, F. (2024). *Machine Learning Robustness: A Primer* (Version 3). https://doi.org/10.48550/ARXIV.2404.00897

Breiman, L. (2001). Random Forests. *Machine Learning*, *45*, 5–32.

Brown, G., Pocock, A., Zhao, M.-J., & Luján, M. (2012). Conditional Likelihood Maximisation: A Unifying Framework for Information Theoretic Feature Selection. *Journal of Machine Learning Research*, *13*(2), 27–66.

Caparrós, M. G. (2025). Überwachtes und unüberwachtes Lernen. In D. Memmert (Hrsg.), *Künstliche Intelligenz und maschinelles Lernen in der Sportwissenschaft* (S. 35–46). Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-662-68950-9_2

Cattell, R. (2010). Scalable SQL and NoSQL Data Stores. *SIGMOD Record*, *39*(4), 12–27. https://doi.org/10.1145/1978915.1978919

Chandrashekar, G., & Sahin, F. (2014). A Survey on Feature Selection Methods. *Computers & Electrical Engineering*, *40*, 16–28. https://doi.org/10.1016/j.compeleceng.2013.11.024

Chapman, D. W. (1976). A Survey of Nonresponse Imputation Procedures. *Proceedings of the Social Statistics Section*, 245–251.

Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: Synthetic Minority Over-sampling Technique. *Journal of Artificial Intelligence Research*, *16*, 321–357. https://doi.org/10.1613/jair.953

Cleff, T. (2015). *Deskriptive Statistik und explorative Datenanalyse: Eine computergestützte Einführung mit Excel, SPSS Und Stata* (3. Aufl.). Springer Gabler Wiesbaden. https://doi.org/10.1007/978-3-8349-4748-2

Cover, T. M., & Thomas, J. A. (2005). *Elements of Information Theory*. Wiley. https://doi.org/10.1002/047174882X

Elmasri, R., & Navathe, S. (2016). *Fundamentals of Database Systems* (7. Aufl.). Pearson.

Emmanuel, T., Maupong, T., Mpoeleng, D., Semong, T., Mphago, B., & Tabona, O. (2021). A Survey on Missing Data in Machine Learning. *Journal of Big Data*, *8*, 140. https://doi.org/10.1186/s40537-021-00516-9

Evans, R. W., Cooley, P. C., & Piserchia, P. V. (1979). A Test for Evaluating Missing Data Imputation Procedures. *Proceedings of the Social Statistics Section*, 469–474.

Fawcett, T. (2006). An Introduction to ROC Analysis. *Pattern Recognition Letters*, *27*(8), 861–874. https://doi.org/10.1016/j.patrec.2005.10.010

Fernstad, S. J. (2019). To Identify What Is Not There: A Definition of Missingness Patterns and Evaluation of Missing Value Visualization. *Information Visualization*, *18*(2), 230–250. https://doi.org/10.1177/1473871618785387

Fleck, P., Kügel, M., & Kommenda, M. (2020). Understanding and Preparing Data of Industrial Processes for Machine Learning Applications. In R. Moreno-Díaz, F. Pichler, & A. Quesada-Arencibia (Hrsg.), *Computer Aided Systems Theory – EUROCAST 2019* (Bd. 12013, S. 413–420). Springer International Publishing. https://doi.org/10.1007/978-3-030-45093-9_50

Ford, B. L. (1983). An Overview of Hot-Deck Procedures. In W. G. Madow, H. Nisselson, & I. Olkin (Hrsg.), *Incomplete Data in Sample Surveys: Bd. Theory and Bibliographies* (S. 185–207). Academic Press.

Freiesleben, T., & Grote, T. (2023). Beyond Generalization: A Theory of Robustness in Machine Learning. *Synthese*, *202*(4), 109. https://doi.org/10.1007/s11229-023-04334-9

Friedman, J. H. (2001). Greedy Function Approximation: A Gradient Boosting Machine. *The Annals of Statistics*, *29*(5), 1189–1232. https://doi.org/10.1214/aos/1013203451

Garcia, S., Derrac, J., Cano, J., & Herrera, F. (2012). Prototype Selection for Nearest Neighbor Classification: Taxonomy and Empirical Study. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *34*(3), 417–435. https://doi.org/10.1109/TPAMI.2011.142

García, S., Luengo, J., & Herrera, F. (2015). *Data Preprocessing in Data Mining* (Bd. 72). Springer International Publishing. https://doi.org/10.1007/978-3-319-10247-4

Gewers, F. L., Ferreira, G. R., Arruda, H. F. de, Silva, F. N., Comin, C. H., Amancio, D. R., & Costa, L. da F. (2022). Principal Component Analysis: A Natural Approach to Data Exploration. *ACM Computing Surveys*, *54*(4), 1–34. https://doi.org/10.1145/3447755

Gomes, H. M., Bifet, A., Read, J., Barddal, J. P., Enembreck, F., Pfahringer, B., Holmes, G., & Abdessalem, T. (2017). Adaptive Random Forests for Evolving Data Stream Classification. *Machine Learning*, *106*(9), 1469–1495. https://doi.org/10.1007/s10994-017-5642-8

Guyon, I., & Elisseeff, A. (2003). An Introduction to Variable and Feature Selection. *Journal of Machine Learning Research*, *3*, 1157–1182.

Guyon, I., Weston, J., Barnhill, S., & Vapnik, V. (2002). Gene Selection for Cancer Classification Using Support Vector Machines. *Machine Learning*, *46*, 389–422. https://doi.org/10.1023/A:1012487302797

Hall, M. A. (1999). *Correlation-Based Feature Selection for Machine Learning*. University of Waikato.

Han, J., Kamber, M., & Pei, J. (2012). *Data Mining: Concepts and Techniques* (3. Aufl.). Morgan Kaufman.

Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The Elements of Statistical Learning*. Springer New York. https://doi.org/10.1007/978-0-387-84858-7

Hedderich, J., & Sachs, L. (2016). *Angewandte Statistik*. Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-662-45691-0

Hodge, V., & Austin, J. (2004). A Survey of Outlier Detection Methodologies. *Artificial Intelligence Review*, *22*(2), 85–126. https://doi.org/10.1023/B:AIRE.0000045502.10941.a9

Hossin, M., & Sulaiman, M. N. (2015). A Review on Evaluation Metrics for Data Classification Evaluations. *International Journal of Data Mining & Knowledge Management Process*, *5*(2), 1–11. https://doi.org/10.5121/ijdkp.2015.5201

imbalanced-learn developers. (2023). *Imbalanced-learn* (Version 0.11.0) [Software]. https://imbalanced-learn.org/stable/

Jahangiri, M., Kazemnejad, A., Goldfeld, K. S., Daneshpour, M. S., Mostafaei, S., Khalili, D., Moghadas, M. R., & Akbarzadeh, M. (2023). A Wide Range of Missing Imputation Approaches in Longitudinal Data: A Simulation Study and Real Data Analysis. *BMC Medical Research Methodology*, *23*, 161. https://doi.org/10.1186/s12874-023-01968-8

James, G., Witten, D., Hastie, T., Tibshirani, R., & Taylor, J. (2023). Linear Regression. In *An Introduction to Statistical Learning: With Applications in Python* (S. 69–134). Springer International Publishing. https://doi.org/10.1007/978-3-031-38747-0_3

Jerez, J. M., Molina, I., García-Laencina, P. J., Alba, E., Ribelles, N., Martín, M., & Franco, L. (2010). Missing Data Imputation Using Statistical and Machine Learning Methods in a real Breast Cancer Problem. *Artificial Intelligence in Medicine*, *50*(2), 105–115. https://doi.org/10.1016/j.artmed.2010.05.002

Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q., & Liu, T.-Y. (2017). LightGBM: A Highly Efficient Gradient Boosting Decision Tree. *Advances in Neural Information Processing Systems*, *30*, 3149–3157.

Kim, J. H. (2019). Multicollinearity and Misleading Statistical Results. *Korean Journal of Anesthesiology*, *72*(6), 558–569. https://doi.org/10.4097/kja.19087

Kohavi, R. (1995). A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection. *Proceedings of the 14th International Joint Conference on Artificial Intelligence - Volume 2*, 1137–1143.

Kohavi, R., & John, G. H. (1997). Wrappers for Feature Subset Selection. *Artificial Intelligence*, *97*, 273–324. https://doi.org/10.1016/S0004-3702(97)00043-X

Krcmar, H. (2015). *Informationsmanagement*. Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-662-45863-1

Kuzudisli, C., Bakir-Gungor, B., Bulut, N., Qaqish, B., & Yousef, M. (2023). Review of feature selection approaches based on grouping of features. *PeerJ*, *11*. https://doi.org/10.7717/peerj.15666

Kyriazos, T., & Poga, M. (2023). Dealing with Multicollinearity in Factor Analysis: The Problem, Detections, and Solutions. *Open Journal of Statistics*, *13*(03), 404–424. https://doi.org/10.4236/ojs.2023.133020

Lall, R., & Robinson, T. (2022). The MIDAS Touch: Accurate and Scalable Missing-Data Imputation with Deep Learning. *Political Analysis*, *30*(2), 179–196. Cambridge Core. https://doi.org/10.1017/pan.2020.49

Lanquillon, C. (2019). Grundzüge des maschinellen Lernens. In S. Schacht & C. Lanquillon (Hrsg.), *Blockchain und maschinelles Lernen: Wie das maschinelle Lernen und die Distributed-Ledger-Technologie voneinander profitieren* (S. 89–142). Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-662-60408-3_3

Leevy, J. L., Khoshgoftaar, T. M., Bauder, R. A., & Seliya, N. (2018). A Survey on Addressing High-Class Imbalance in Big Data. *Journal of Big Data*, *5*, 42. https://doi.org/10.1186/s40537-018-0151-6

Li, J., Cheng, K., Wang, S., Morstatter, F., Trevino, R. P., Tang, J., & Liu, H. (2017). Feature Selection: A Data Perspective. *ACM Computing Surveys*, *50*(6), 1–45. https://doi.org/10.1145/3136625

Little, R. (1988). A Test of Missing Completely at Random for Multivariate Data with Missing Values. *Journal of the American Statistical Association*, *83*(404), 1198–1202. https://doi.org/10.1080/01621459.1988.10478722

Little, R., & Rubin, D. (2019). *Statistical Analysis with Missing Data* (3. Aufl.). Wiley. https://doi.org/10.1002/9781119482260

Liu, H., & Motoda, H. (Hrsg.). (2007). *Computational Methods of Feature Selection*. Chapman and Hall/CRC. https://doi.org/10.1201/9781584888796

Liu, H., & Yu, L. (2005). Toward Integrating Feature Selection Algorithms for Classification and Clustering. *IEEE Transactions on Knowledge and Data Engineering*, *17*(4), 491–502. https://doi.org/10.1109/TKDE.2005.66

Madow, W. G., Nisselson, H., & Olkin, I. (Hrsg.). (1983). *Incomplete Data in Sample Surveys: Bd. Report and Case Studies*. Academic Press.

Mahlich, C., Vente, T., & Beel, J. (2024). *From Theory to Practice: Implementing and Evaluating E-Fold Cross-Validation*. https://doi.org/10.48550/arXiv.2410.09463

Manning, C. D., Raghavan, P., & Schütze, H. (2008). *Introduction to Information Retrieval*. Cambridge University Press.

McCann, M., & Johnston, A. (2008). *SECOM* [Dataset]. UCI Machine Learning Repository. https://doi.org/10.24432/C54305

Neuer, M. J. (2024). Datenvorverarbeitung. In M. J. Neuer, *Maschinelles Lernen für die Ingenieurwissenschaften* (S. 57–91). Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-662-68216-6_3

North, K. (2021). *Wissensorientierte Unternehmensführung: Wissensmanagement im digitalen Wandel*. Springer Fachmedien Wiesbaden. https://doi.org/10.1007/978-3-658-32771-2

NumPy developers. (2023). *NumPy* (Version 1.25.2) [Software]. https://numpy.org/doc/1.25/

O'Brien, R. M. (2007). A Caution Regarding Rules of Thumb for Variance Inflation Factors. *Quality & Quantity*, *41*(5), 673–690. https://doi.org/10.1007/s11135-006-9018-6

Olvera-López, J. A., J. Ariel Carrasco-Ochoa, Martínez-Trinidad, J. F., & Kittler, J. (2010). A Review of Instance Selection Methods. *Artificial Intelligence Review*, *34*(2), 133–143. https://doi.org/10.1007/s10462-010-9165-y

pandas development team. (2025). *Pandas* (Version 2.3.1) [Software]. https://pandas.pydata.org/docs/

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, É. (2011). Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, *12*(85), 2825–2830.

Pereira, R. C., Santos, M. S., Rodrigues, P. P., & Abreu, P. H. (2020). Reviewing Autoencoders for Missing Data Imputation: Technical Trends, Applications and Outcomes. *Journal of Artificial Intelligence Research*, *69*, 1255–1285. https://doi.org/10.1613/jair.1.12312

Prakash, P., Street, K., Narayanan, S., Fernandez, B. A., Shen, Y., & Shu, C. (2024). Benchmarking Machine Learning Missing Data Imputation Methods in Large-Scale Mental Health Survey Databases. *Artificial Intelligence in Health*, *2*, 81. https://doi.org/10.36922/aih.4406

Python Software Foundation. (2023). *Python* (Version 3.11.5) [Software]. https://docs.python.org/release/3.11.5/

Radovic, M., Ghalwash, M., Filipovic, N., & Obradovic, Z. (2017). Minimum Redundancy Maximum Relevance Feature Selection Approach for Temporal Gene Expression Data. *BMC Bioinformatics*, *18*, 9. https://doi.org/10.1186/s12859-016-1423-9

Rahm, E., & Do, H. H. (2000). Data Cleaning: Problems and Current Approaches. *IEEE Bulletin on Data Engineering*, *23*(4), 3–13.

Ranawana, R., & Palade, V. (2006). Optimized Precision—A New Measure for Classifier Performance Evaluation. *2006 IEEE International Conference on Evolutionary Computation*, 2254–2261. https://doi.org/10.1109/CEC.2006.1688586

Rebstadt, J., Kortum, H., Gravemeier, L. S., Eberhardt, B., & Thomas, O. (2022). Non-Discrimination-by-Design: Handlungsempfehlungen für die Entwicklung von vertrauenswürdigen KI-Services. *HMD Praxis der Wirtschaftsinformatik*, *59*(2), 495–511. https://doi.org/10.1365/s40702-022-00847-y

Rodrigues, D., & Da Silva, A. (2021). A Study on Machine Learning Techniques for the Schema Matching Network Problem. *Journal of the Brazilian Computer Society*, *27*, 14. https://doi.org/10.1186/s13173-021-00119-5

Saeys, Y., Inza, I., & Larrañaga, P. (2007). A Review of Feature Selection Techniques in Bioinformatics. *Bioinformatics*, *23*(19), 2507–2517. https://doi.org/10.1093/bioinformatics/btm344

Saporta, G., & Niang, N. (2009). Principal Component Analysis: Application to Statistical Process Control. In *Data Analysis* (S. 1–23). https://doi.org/10.1002/9780470611777.ch1

Schafer, J. L., & Graham, J. W. (2002). Missing Data: Our View of the State of the Art. *Psychological Methods*, *7*(2), 147–177. https://doi.org/10.1037//1082-989x.7.2.147

Schicker, E. (2017). *Datenbanken und SQL*. Springer Fachmedien Wiesbaden. https://doi.org/10.1007/978-3-658-16129-3

Schnell, R. (1986). *Missing-Data-Probleme in der Empirischen Sozialforschung*. Ruhr-Universität Bochum.

Schwartz, R., Dodge, J., Smith, N. A., & Etzioni, O. (2020). Green AI. *Communications of the ACM*, *63*(12), 54–63. https://doi.org/10.1145/3381831

scikit-learn developers. (2023). *Scikit-learn* (Version 1.3.2) [Software]. https://scikit-learn.org/1.3/preface.html

statsmodels developers. (2023). *Statsmodels* (Version 0.14.0) [Software]. https://www.statsmodels.org/v0.14.0/user-guide.html

Stekhoven, D. J., & Bühlmann, P. (2012). MissForest—Non-Parametric Missing Value Imputation for Mixed-Type Data. *Bioinformatics*, *28*, 112–118. https://doi.org/10.1093/bioinformatics/btr597

Strubell, E., Ganesh, A., & McCallum, A. (2019). Energy and Policy Considerations for Deep Learning in NLP. *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 3645–3650. https://doi.org/10.48550/ARXIV.1906.02243

Tan, P.-N., Karpatne, A., Steinbach, M., & Kumar, V. (2019). *Introduction to Data Mining: Global Edition*. Pearson Deutschland.

Teh, H. Y., Kempa-Liehr, A. W., & Wang, K. I.-K. (2020). Sensor Data Quality: A Systematic Review. *Journal of Big Data*, *7*, 11. https://doi.org/10.1186/s40537-020-0285-1

Theng, D., & Bhoyar, K. K. (2024). Feature Selection Techniques for Machine Learning: A Survey of More Than Two Decades of Research. *Knowledge and Information Systems*, *66*(3), 1575–1637. https://doi.org/10.1007/s10115-023-02010-5

Tibshirani, R. (1996). Regression Shrinkage and Selection via the Lasso. *Journal of the Royal Statistical Society, Series B (Methodological)*, *58*, 267–288. JSTOR.

Tierney, N., & Cook, D. (2023). Expanding Tidy Data Principles to Facilitate Missing Data Exploration, Visualization and Assessment of Imputations. *Journal of Statistical Software*, *105*(7). https://doi.org/10.18637/jss.v105.i07

Triguero, I., Peralta, D., Bacardit, J., García, S., & Herrera, F. (2015). MRPR: A MapReduce Solution for Prototype Reduction in Big Data Classification. *Neurocomputing*, *150*, 331–345. https://doi.org/10.1016/j.neucom.2014.04.078

van Buuren, S. (2018). *Flexible Imputation of Missing Data* (2. Aufl.). Chapman and Hall/CRC. https://doi.org/10.1201/9780429492259

Vergara, J. R., & Estévez, P. A. (2014). A Review of Feature Selection Methods Based on Mutual Information. *Neural Computing and Applications*, *24*, 175–186. https://doi.org/10.1007/s00521-013-1368-0

von Enzberg, S., Weller, J., Brock, J., Merkelbach, S., Panzner, M., Lick, J., Kühn, A., & Dumitrescu, R. (2024). On the Current State of Industrial Data Science: Challenges, Best Practices, and Future Directions. *Procedia CIRP*, *130*, 1454–1461. https://doi.org/10.1016/j.procir.2024.10.266

Wang, Z., Akande, O., Poulos, J., & Li, F. (2021). *Are Deep Learning Models Superior for Missing Data Imputation in Large Surveys? Evidence from an Empirical Comparison* (Version 3). arXiv. https://doi.org/10.48550/ARXIV.2103.09316

Wilson, D. R., & Martinez, T. R. (2000). Reduction Techniques for Instance-Based Learning Algorithms. *Machine Learning*, *38*(3), 257–286. https://doi.org/10.1023/A:1007626913721

Wishart, D. (1978). Treatment of Missing Values in Cluster Analysis. *Proceedings in Computational Statistics (COMPSTAT 1978)*, 281–297.

Witten, I. H., Frank, E., Hall, M. A., & Pal, C. J. (2017). *Data Mining: Practical Machine Learning Tools and Techniques* (4. Aufl.). Elsevier, Morgan Kaufmann.

Zeigermann, O., & Nguyen, C. N. (2024). *Machine Learning kurz & gut* (3. Aufl.). O'Reilly Verlag.

Zhang, Z. (2016). Introduction to Machine Learning: K-Nearest Neighbors. *Annals of Translational Medicine*, *4*(11). https://doi.org/10.21037/atm.2016.03.37

Anhang

Anhang A: Übersicht der implementierten Versuchskonfigurationen

Tabelle A1: Versuchsübersicht des Experimentierplans

Versuch	DVV 1	DVV 2	Versuchstyp
V01	Mittelwertimputation	–	Einzelverfahren
V02	Medianimputation	–	Einzelverfahren
V03	Konstantenimputation	–	Einzelverfahren
V04	kNN-Imputation	–	Einzelverfahren
V05	Iterative Imputation	–	Einzelverfahren
V06	CFS	–	Einzelverfahren
V07	MI	–	Einzelverfahren
V08	VIF	–	Einzelverfahren
V09	PCA	–	Einzelverfahren
V10	ENN	–	Einzelverfahren
V11	Mittelwertimputation	CFS	Kombination
V12	CFS	Mittelwertimputation	Kombination
V13	Mittelwertimputation	MI	Kombination
V14	MI	Mittelwertimputation	Kombination
V15	Mittelwertimputation	VIF	Kombination
V16	VIF	Mittelwertimputation	Kombination
V17	Mittelwertimputation	PCA	Kombination
V18	PCA	Mittelwertimputation	Kombination
V19	Mittelwertimputation	ENN	Kombination
V20	ENN	Mittelwertimputation	Kombination
V21	Medianimputation	CFS	Kombination
V22	CFS	Medianimputation	Kombination
V23	Medianimputation	MI	Kombination
V24	MI	Medianimputation	Kombination
V25	Medianimputation	VIF	Kombination
V26	VIF	Medianimputation	Kombination
V27	Medianimputation	PCA	Kombination
V28	PCA	Medianimputation	Kombination
V29	Medianimputation	ENN	Kombination
V30	ENN	Medianimputation	Kombination
V31	Konstantenimputation	CFS	Kombination
V32	CFS	Konstantenimputation	Kombination
V33	Konstantenimputation	MI	Kombination
V34	MI	Konstantenimputation	Kombination

Versuch	DVV 1	DVV 2	Versuchstyp
V35	Konstantenimputation	VIF	Kombination
V36	VIF	Konstantenimputation	Kombination
V37	Konstantenimputation	PCA	Kombination
V38	PCA	Konstantenimputation	Kombination
V39	Konstantenimputation	ENN	Kombination
V40	ENN	Konstantenimputation	Kombination
V41	kNN-Imputation	CFS	Kombination
V42	CFS	kNN-Imputation	Kombination
V43	kNN-Imputation	MI	Kombination
V44	MI	kNN-Imputation	Kombination
V45	kNN-Imputation	VIF	Kombination
V46	VIF	kNN-Imputation	Kombination
V47	kNN-Imputation	PCA	Kombination
V48	PCA	kNN-Imputation	Kombination
V49	kNN-Imputation	ENN	Kombination
V50	ENN	kNN-Imputation	Kombination
V51	Iterative Imputation	CFS	Kombination
V52	CFS	Iterative Imputation	Kombination
V53	Iterative Imputation	MI	Kombination
V54	MI	Iterative Imputation	Kombination
V55	Iterative Imputation	VIF	Kombination
V56	VIF	Iterative Imputation	Kombination
V57	Iterative Imputation	PCA	Kombination
V58	PCA	Iterative Imputation	Kombination
V59	Iterative Imputation	ENN	Kombination
V60	ENN	Iterative Imputation	Kombination

Anhang B: Übersicht der Versuchsergebnisse

Tabelle B1: Versuchsergebnisse nach Ausführung der Experimente

Versuch	Accuracy (MW)	Accuracy (SD)	Precision (MW)	Precision (SD)	Recall (MW)	Recall (SD)	F1 (MW)	F1 (SD)	Zeit (s)
V01	0,8630	0,1009	0,8102	0,1316	0,9836	0,0232	0,8840	0,0840	9,2
V02	0,8739	0,1025	0,8250	0,1305	0,9816	0,0261	0,8924	0,0853	9,1
V03	0,8733	0,1091	0,8262	0,1375	0,9822	0,0251	0,8928	0,0897	9,3
V04	0,8770	0,1040	0,8286	0,1334	0,9857	0,0203	0,8957	0,0855	10,8
V05	0,8691	0,1033	0,8183	0,1332	0,9850	0,0213	0,8893	0,0852	428,8
V06	0,8418	0,1146	0,7894	0,1439	0,9788	0,0299	0,8685	0,0939	3,3
V07	0,8384	0,0907	0,7709	0,1009	0,9945	0,0063	0,8651	0,0672	4,0
V08	0,8685	0,0946	0,8134	0,1180	0,9857	0,0189	0,8877	0,0770	4,1
V09	0,8257	0,0627	0,7691	0,0763	0,9453	0,0288	0,8465	0,0512	1,9
V10	0,8716	0,0977	0,8209	0,1296	0,9822	0,0252	0,8901	0,0827	10,5
V11	0,8753	0,1110	0,8297	0,1364	0,9802	0,0266	0,8942	0,0907	4,4
V12	0,8230	0,1188	0,7687	0,1475	0,9802	0,0266	0,8554	0,0955	2,8
V13	0,8175	0,0948	0,7481	0,1070	0,9966	0,0048	0,8507	0,0701	3,9
V14	0,8493	0,0942	0,7843	0,1048	0,9966	0,0048	0,8741	0,0694	3,8
V15	0,8582	0,1019	0,8041	0,1280	0,9816	0,0261	0,8798	0,0839	59,8
V16	0,8849	0,0727	0,8296	0,0873	0,9822	0,0237	0,8979	0,0607	4,1
V17	0,8363	0,0230	0,7934	0,0289	0,9111	0,0126	0,8480	0,0192	2,0
V18	0,8257	0,0627	0,7691	0,0763	0,9453	0,0288	0,8465	0,0512	2,1
V19	0,8679	0,1034	0,8162	0,1364	0,9840	0,0226	0,8874	0,0867	7,7
V20	0,8885	0,1079	0,8399	0,1440	0,9885	0,0065	0,9017	0,0898	7,6
V21	0,8736	0,1063	0,8250	0,1333	0,9836	0,0232	0,8929	0,0870	4,8
V22	0,8145	0,1171	0,7591	0,1438	0,9781	0,0266	0,8487	0,0933	3,1
V23	0,8152	0,1023	0,7481	0,1104	0,9925	0,0092	0,8490	0,0744	3,8
V24	0,8292	0,1068	0,7634	0,1094	0,9959	0,0058	0,8601	0,0752	4,2

Versuch	Accuracy (MW)	Accuracy (SD)	Precision (MW)	Precision (SD)	Recall (MW)	Recall (SD)	F1 (MW)	F1 (SD)	Zeit (s)
V25	0,8671	0,0983	0,8147	0,1219	0,9788	0,0299	0,8858	0,0818	73,3
V26	0,8773	0,0829	0,8226	0,1025	0,9816	0,0261	0,8927	0,0695	3,6
V27	0,8339	0,0227	0,7973	0,0311	0,8975	0,0044	0,8442	0,0187	1,8
V28	0,8257	0,0627	0,7691	0,0763	0,9453	0,0288	0,8465	0,0512	2,3
V29	0,8759	0,1050	0,8280	0,1349	0,9805	0,0261	0,8933	0,0878	7,8
V30	0,8853	0,1037	0,8334	0,1418	0,9900	0,0044	0,8986	0,0869	8,1
V31	0,8685	0,1125	0,8207	0,1378	0,9809	0,0270	0,8890	0,0915	4,5
V32	0,8165	0,1209	0,7637	0,1476	0,9720	0,0396	0,8493	0,0981	3,1
V33	0,8141	0,0976	0,7449	0,0993	0,9891	0,0155	0,8468	0,0706	3,9
V34	0,8049	0,0938	0,7347	0,1019	0,9918	0,0089	0,8406	0,0689	4,1
V35	0,8897	0,1343	0,8604	0,1565	0,9720	0,0382	0,9077	0,1090	111,6
V36	0,8951	0,0785	0,8486	0,0957	0,9754	0,0319	0,9060	0,0675	4,8
V37	0,8442	0,0218	0,7986	0,0304	0,9228	0,0151	0,8558	0,0179	2,1
V38	0,8257	0,0627	0,7691	0,0763	0,9453	0,0288	0,8465	0,0512	2,0
V39	0,8750	0,1108	0,8276	0,1379	0,9819	0,0255	0,8935	0,0910	7,7
V40	0,8831	0,0978	0,8290	0,1380	0,9892	0,0079	0,8960	0,0828	7,0
V41	0,8756	0,1095	0,8281	0,1367	0,9863	0,0193	0,8954	0,0885	6,1
V42	0,8237	0,1142	0,7678	0,1432	0,9822	0,0209	0,8558	0,0917	3,3
V43	0,8445	0,1030	0,7843	0,1153	0,9857	0,0174	0,8698	0,0786	5,2
V44	0,8251	0,0990	0,7585	0,1026	0,9891	0,0102	0,8552	0,0713	4,2
V45	0,8743	0,1052	0,8246	0,1273	0,9836	0,0232	0,8931	0,0850	79,5
V46	0,8801	0,0707	0,8251	0,0877	0,9781	0,0251	0,8935	0,0600	3,6
V47	0,8301	0,0237	0,7907	0,0361	0,9016	0,0074	0,8420	0,0183	3,3
V48	0,8257	0,0627	0,7691	0,0763	0,9453	0,0288	0,8465	0,0512	1,7

Versuch	Accuracy (MW)	Accuracy (SD)	Precision (MW)	Precision (SD)	Recall (MW)	Recall (SD)	F1 (MW)	F1 (SD)	Zeit (s)
V49	0,8769	0,1080	0,8293	0,1364	0,9826	0,0246	0,8948	0,0894	8,6
V50	0,8853	0,1130	0,8354	0,1483	0,9931	0,0033	0,9003	0,0924	11,7
V51	0,8688	0,1084	0,8190	0,1345	0,9843	0,0222	0,8894	0,0879	437,2
V52	0,8192	0,1165	0,7636	0,1463	0,9843	0,0194	0,8533	0,0924	6,6
V53	0,8141	0,1042	0,7487	0,1113	0,9891	0,0068	0,8480	0,0751	446,2
V54	0,8240	0,0878	0,7530	0,0953	0,9952	0,0068	0,8542	0,0642	6,4
V55	0,8541	0,1088	0,8006	0,1334	0,9822	0,0251	0,8775	0,0880	516,1
V56	0,8750	0,0922	0,8211	0,1079	0,9822	0,0237	0,8917	0,0745	4,9
V57	0,8305	0,0220	0,7908	0,0328	0,9016	0,0030	0,8421	0,0170	455,5
V58	0,8257	0,0627	0,7691	0,0763	0,9453	0,0288	0,8465	0,0512	3,6
V59	0,8732	0,1043	0,8234	0,1353	0,9847	0,0202	0,8920	0,0863	452,9
V60	0,8885	0,1043	0,8386	0,1411	0,9900	0,0029	0,9016	0,0865	632,7

Anhang C: Ergebnisdiagramme Medianimputation und kNN-Imputation

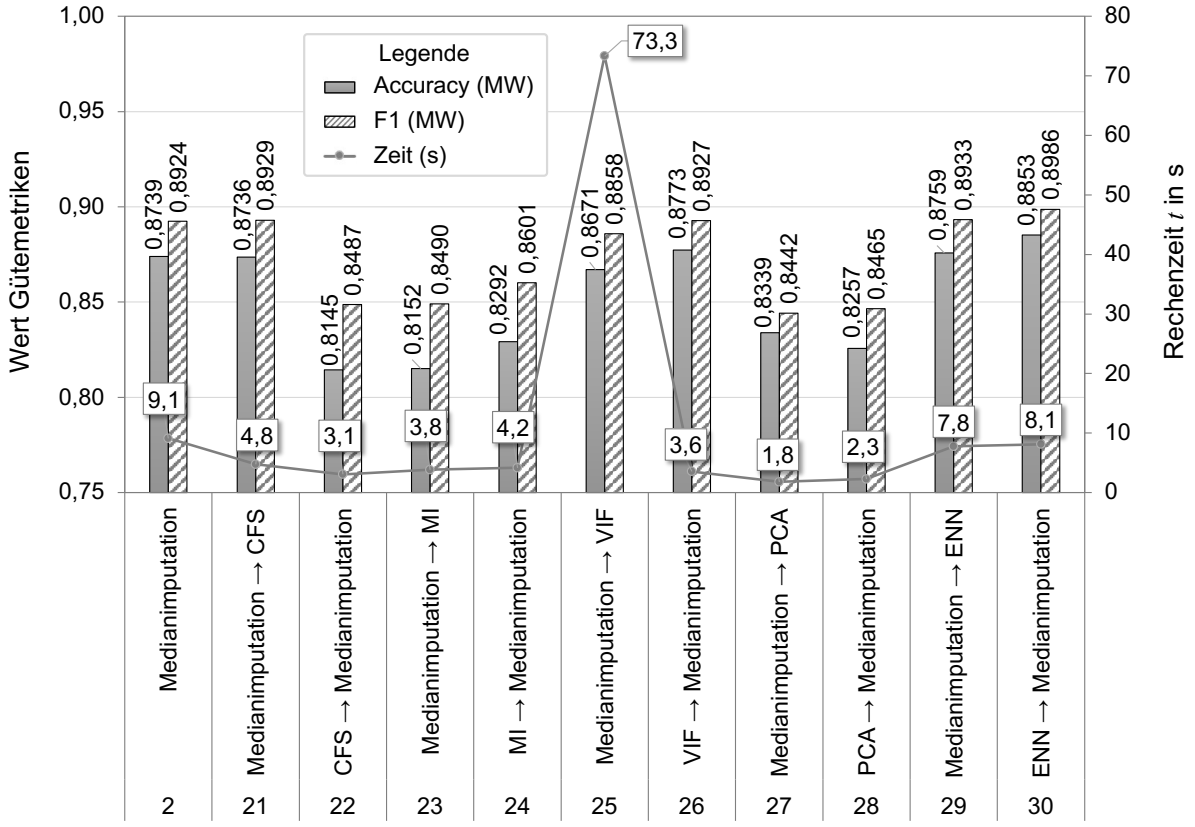


Abbildung C1: Versuchsergebnisse der Verfahrenskombinationen mit der Medianimputation

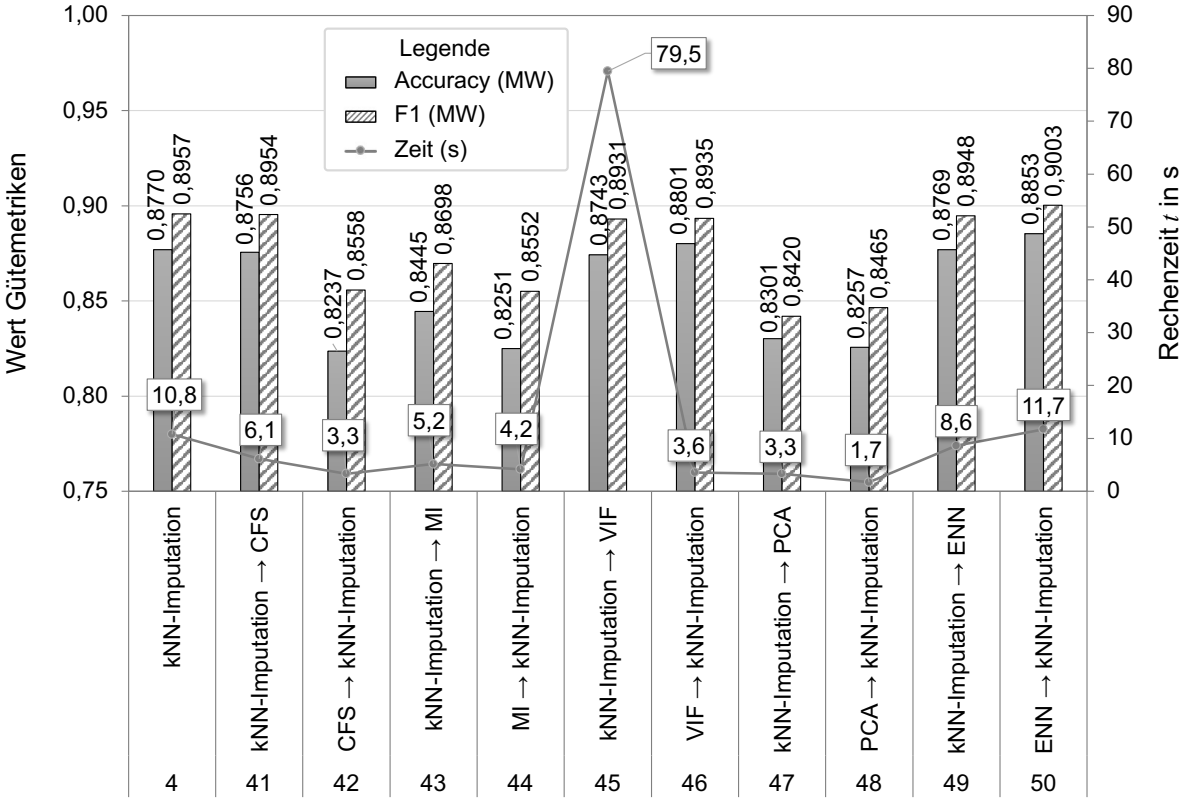


Abbildung C2: Versuchsergebnisse der Verfahrenskombinationen mit der kNN-Imputation

Anhang D: Quellcode

```
import pandas as pd
import numpy as np
import time
from sklearn.experimental import enable_iterative_imputer
from statsmodels.stats.outliers_influence import variance_inflation_factor
from sklearn.ensemble import HistGradientBoostingClassifier
from sklearn.impute import SimpleImputer, KNNImputer, IterativeImputer
from sklearn.feature_selection import mutual_info_classif
from sklearn.decomposition import PCA
from sklearn.model_selection import cross_validate
from sklearn.metrics import make_scorer, accuracy_score, precision_score, recall_score, f1_score
from imblearn.over_sampling import SMOTE
from imblearn.under_sampling import EditedNearestNeighbours

# Sammelliste für die Ergebnisübersicht
summary_rows = []

# 1. Daten einlesen
pfad = "Hier Pfad zum Datensatzordner einfügen"
X_raw = pd.read_csv(pfad + "secom.data", sep=" ", header=None).dropna(axis=1, how='all') # Features laden; komplett
leere Spalten entfernen
y_raw = pd.read_csv(pfad + "secom_labels.data", sep=" ", header=None).iloc[:, 0] # Label-Spalte laden

# 2. Modell & Evaluationsmetriken definieren
model = HistGradientBoostingClassifier(random_state=42) # Modellinstanz mit Fixierung des Zufallswertes
scoring = { # Dictionary der zu berechnenden Metriken (zero_division=0: 0 vermeiden bei Division)
    "accuracy": make_scorer(accuracy_score),
    "precision": make_scorer(precision_score, zero_division=0),
    "recall": make_scorer(recall_score, zero_division=0),
    "f1": make_scorer(f1_score, zero_division=0),
}

def evaluate_model(X, y): # Hilfsfunktion für Kreuzvalidierung (CV)
    return cross_validate(model, X, y, cv=3, scoring=scoring) # 3-fold CV mit obigem Modell/Metriken

def _add_row(name, category, results, elapsed, meta=None): # Ergebniszeile zusammenbauen und anhängen
    meta = meta or {}
    row = { # Dictionary mit Kennzahlen und Meta
        "Versuch": name, # Versuchsbezeichnung
        "Kategorie": category, # Baseline / Imputation / Reduktion / Kombination
        "Accuracy_MW": float(results["test_accuracy"].mean()),
        "Accuracy_SD": float(results["test_accuracy"].std()),
        "Precision_MW": float(results["test_precision"].mean()),
        "Precision_SD": float(results["test_precision"].std()),
        "Recall_MW": float(results["test_recall"].mean()),
        "Recall_SD": float(results["test_recall"].std()),
        "F1_MW": float(results["test_f1"].mean()),
        "F1_SD": float(results["test_f1"].std()),
        "Zeit_s": float(elapsed)
    }
    summary_rows.append(row) # Zeile zur Gesamtliste hinzufügen

# 3. CFS, MI & VIF als Funktionen definieren
def apply_cfs(X, y): # Korrelation-basiertes Feature-Subset (vereinfachte CFS-Variante)
    corrs = X.apply(lambda col: np.abs(np.corrcoef(col, y)[0, 1])) # Korrelation jedes Features mit Zielvariable
berechnen
    corr_matrix = X.corr().abs() # Korrelation zwischen allen Features
    selected = [] # Hier landen die ausgewählten Features
    for col in corrs.sort_values(ascending=False).index: # Iterieren durch die Features in absteigender Reihenfolge
(stärkste Korrelation mit Ziel zuerst)
        if not selected or all(corr_matrix[col][selected] < 0.7): # Feauture nur wählen, wenn es zu allen bereits
gewählten Features eine geringe Korrelation (< 0.7) hat
            selected.append(col) # Feature hinzufügen
    return X[selected] # DataFrame mit den ausgewählten Spalten

def apply_mi(X, y, k=50): # Mutual Information Top 50
    mi = mutual_info_classif(X, y.values.ravel()) # MI-Werte pro Feature
    top_k_idx = np.argsort(mi)[-k:] # Indizes der k größten MI-Werte
    return X.iloc[:, top_k_idx] # Spalten-Subset anhand Indizes

def apply_vif(X): # VIF-basierte Feature-Selektion
    corr_matrix = X.corr().abs() # Berechnung der Korrelationsmatrix (absolut, d.h. Vorzeichen egal)
    upper = corr_matrix.where(np.triu(np.ones(corr_matrix.shape), k=1).astype(bool)) # Betrachtung nur des oberen
Dreiecks der Matrix (sonst doppelte Infos)
    to_drop = [column for column in upper.columns if any(upper[column] > 0.95)] # Markierung aller Spalten, die mit
irgendeiner anderen Spalte sehr hoch korrelieren (> 0.95)
    X = X.drop(columns=to_drop) # Entfernung dieser stark korrelierten Spalten (spart Rechenzeit und reduziert tri-
viale Doppelungen)
    vif_data = pd.DataFrame({"feature": X.columns,
                             "VIF": [variance_inflation_factor(X.values, i) for i in range(X.shape[1])]}) # Berech-
nung des VIF für alle verbleibenden Features
    selected = vif_data[vif_data["VIF"] < 5]["feature"].tolist() # nur Features mit VIF < 5 behalten (unterhalb
Schwelle für problematische Multikollinearität)
    return X[selected] # Rückgabe des Teil-DataFrames mit nur den ausgewählten Features

# 4. Verfahren definieren
imputation_methods = { # Sammlung der unterschiedlichen Imputationsverfahren
    "mean": SimpleImputer(strategy="mean"), # Mittelwertimputation
    "median": SimpleImputer(strategy="median"), # Medianimputation
    "constant": SimpleImputer(strategy="constant", fill_value=0), # Konstantenimputation mit 0
    "knn": KNNImputer(),
    "iterative": IterativeImputer(max_iter=3, random_state=42), # Iterative Imputation mit 3 Iterationen und Zu-
fallswert 42
```

```
}
reduction_methods = { # Sammlung der Reduktionsverfahren
    "cfs": "cfs", # Platzhalter-String: ruft apply_cfs auf
    "mi": "mi", # Platzhalter-String: ruft apply_mi auf
    "vif": "vif", # Platzhalter-String: ruft apply_vif auf
    "pca": PCA(n_components=0.95, random_state=42), # PCA mit 95 % Varianz, Zufallswert 42
    "enn": EditedNearestNeighbours() # ENN-Verfahren (Instanzenreduktion)
}

# 5. Baseline

# Baseline ohne Vorverarbeitung
start = time.time() # Startzeit messen
results = evaluate_model(X_raw, y_raw) # CV auf Rohdaten
_add_row("Baseline (ohne Vorverarbeitung)", "Baseline", results, time.time() - start) # Ergebnisse protokollieren

# Baseline mit Dummy-Imputation + SMOTE + Dummys wieder entfernen
start = time.time() # Startzeit messen
X_dummy = SimpleImputer(strategy="constant", fill_value=999999).fit_transform(X_raw) # Dummyimputation
X_bal, y_bal = SMOTE(random_state=42).fit_resample(X_dummy, y_raw) # SMOTE (Balance herstellen)
X_bal = pd.DataFrame(X_bal).replace(999999, np.nan) # Dummywerte zurück auf NaN setzen
results = evaluate_model(X_bal, y_bal) # CV anwenden
_add_row("Baseline (Dummy-Imputation + SMOTE)", "Baseline", results, time.time() - start) # Ergebnisse protokollie-
ren

# 6. Einzelverfahren

# Imputationsverfahren
for name, imputer in imputation_methods.items(): # über alle Imputationsverfahren iterieren
    start = time.time() # Startzeit messen
    X_imp = imputer.fit_transform(X_raw) # Imputation durchführen
    X_imp, y_imp = SMOTE(random_state=42).fit_resample(X_imp, y_raw) # SMOTE anwenden
    results = evaluate_model(X_imp, y_imp) # CV ausführen
    _add_row(f"Imputation: {name}", "Imputation", results, time.time() - start) # Ergebnisse protokollieren

# Reduktionsverfahren
for name, reducer in reduction_methods.items(): # über alle Reduktionsverfahren iterieren
    start = time.time()
    X_dummy = SimpleImputer(strategy="constant", fill_value=999999).fit_transform(X_raw) # Dummyimputation für Ver-
fahren, die nicht mit NaN umgehen können
    y_red = y_raw.copy() # Zielvariable kopieren (wird teils überschrieben)

    if name == "cfs": # CFS-Zweig
        X_red = apply_cfs(pd.DataFrame(X_dummy), y_red) # CFS anwenden
    elif name == "mi": # MI-Zweig
        X_red = apply_mi(pd.DataFrame(X_dummy), y_red) # MI anwenden
    elif name == "vif": # VIF-Zweig
        X_red = apply_vif(pd.DataFrame(X_dummy)) # VIF anwenden
    else:
        if hasattr(reducer, "fit_transform"): # z. B. PCA (hat fit_transform); Transformator -> arbeitet spalten-
weise
            X_red = reducer.fit_transform(X_dummy, y_red)
        else:  # Resampling-Verfahren (z. B. ENN); Resampler -> arbeitet zeilenweise
            X_red, y_red = reducer.fit_resample(X_dummy, y_red)

    X_red, y_red = SMOTE(random_state=42).fit_resample(X_red, y_red) # SMOTE anwenden
    X_red = pd.DataFrame(X_red).replace(999999, np.nan) # Dummywerte wieder zu NaN

    results = evaluate_model(X_red, y_red) # CV ausführen
    _add_row(f"Reduktion: {name}", "Reduktion", results, time.time() - start) # Ergebnisse protokollieren

# 7. Verfahrenskombinationen
for imp_name, imputer in imputation_methods.items(): # alle Imputationsverfahren
    for red_name, reducer in reduction_methods.items(): # alle Reduktionsverfahren

        # Imputation → Reduktion
        start = time.time() # Startzeit messen
        X_imp = imputer.fit_transform(X_raw) # zuerst Imputation
        y_res = y_raw.copy() # Zielvariable kopieren

        if red_name == "cfs": # CFS nach Imputation
            X_red = apply_cfs(pd.DataFrame(X_imp), y_res) # CFS anwenden
            X_red, y_res = SMOTE(random_state=42).fit_resample(X_red, y_res) # SMOTE
            y_res = pd.Series(y_res).astype(int) # Zielvariable in Ganzahl (int) umwandeln
        elif red_name == "mi": # MI nach Imputation
            X_red = apply_mi(pd.DataFrame(X_imp), y_res)
            X_red, y_res = SMOTE(random_state=42).fit_resample(X_red, y_res)
            y_res = pd.Series(y_res).astype(int)
        elif red_name == "vif": # VIF nach Imputation
            X_red = apply_vif(pd.DataFrame(X_imp))
            X_red, y_res = SMOTE(random_state=42).fit_resample(X_red, y_res)
            y_res = pd.Series(y_res).astype(int)
        elif red_name == "enn": # ENN nach Imputation
            X_smote, y_smote = SMOTE(random_state=42).fit_resample(X_imp, y_res) # erst SMOTE
            y_smote = pd.Series(y_smote).astype(int)
            X_red, y_res = reducer.fit_resample(X_smote, y_smote) # dann ENN
            y_res = pd.Series(y_res).astype(int)
        else:
            if hasattr(reducer, "fit_transform"): # Merkmalsreduktion/PCA
                X_red = reducer.fit_transform(X_imp, y_res) # Reduzieren
            else:
                X_red, y_res = reducer.fit_resample(X_imp, y_res) # Resampling
                y_res = pd.Series(y_res).astype(int) # Zielvariable in Ganzahl (int) umwandeln
            X_red, y_res = SMOTE(random_state=42).fit_resample(X_red, y_res) # zusätzlich SMOTE
            y_res = pd.Series(y_res).astype(int)

        n_features_out = X_red.shape[1] # Anzahl Features nach Verarbeitung
```

```
        results = evaluate_model(X_red, y_res) # CV ausführen
        _add_row(f"Kombination: {imp_name} → {red_name}", "Kombination", results, time.time() - start) # Ergebnisse
protokollieren

        # Reduktion → Imputation
        start = time.time() # Startzeit
        X_dummy = SimpleImputer(strategy="constant", fill_value=999999).fit_transform(X_raw) # Dummyimputation
        y_res = y_raw.copy() # Zielvariable kopieren

        if red_name == "cfs": # CFS vor Imputation
            X_red = apply_cfs(pd.DataFrame(X_dummy), y_res) # CFS anwenden
            X_red = pd.DataFrame(X_red).replace(999999, np.nan) # Dummy zurück zu NaN
            X_imp2 = imputer.fit_transform(X_red) # dann "richtige" Imputation
            X_imp2, y_res = SMOTE(random_state=42).fit_resample(X_imp2, y_res) # SMOTE anwenden
            y_res = pd.Series(y_res).astype(int) # Zielvariable in Ganzahl (int) umwandeln
            results = evaluate_model(X_imp2, y_res) # CV ausführen
            n_features_out = X_imp2.shape[1] # Anzahl Feautures speichern

        elif red_name == "mi": # MI vor Imputation
            X_red = apply_mi(pd.DataFrame(X_dummy), y_res)
            X_red = pd.DataFrame(X_red).replace(999999, np.nan)
            X_imp2 = imputer.fit_transform(X_red)
            X_imp2, y_res = SMOTE(random_state=42).fit_resample(X_imp2, y_res)
            y_res = pd.Series(y_res).astype(int)
            results = evaluate_model(X_imp2, y_res)
            n_features_out = X_imp2.shape[1]

        elif red_name == "vif": # VIF vor Imputation
            X_red = apply_vif(pd.DataFrame(X_dummy))
            X_red = pd.DataFrame(X_red).replace(999999, np.nan)
            X_imp2 = imputer.fit_transform(X_red)
            X_imp2, y_res = SMOTE(random_state=42).fit_resample(X_imp2, y_res)
            y_res = pd.Series(y_res).astype(int)
            results = evaluate_model(X_imp2, y_res)
            n_features_out = X_imp2.shape[1]

        elif red_name == "enn": # ENN vor Imputation
            X_smote, y_smote = SMOTE(random_state=42).fit_resample(X_dummy, y_res)
            y_smote = pd.Series(y_smote).astype(int)
            X_red, y_res = reducer.fit_resample(X_smote, y_smote)
            y_res = pd.Series(y_res).astype(int)
            X_red = pd.DataFrame(X_red).replace(999999, np.nan)
            X_imp2 = imputer.fit_transform(X_red)
            results = evaluate_model(X_imp2, y_res)
            n_features_out = X_imp2.shape[1]

        else: # z. B. PCA vor Imputation (für Standardisierung des Codes)
            if hasattr(reducer, "fit_transform"): # Merkmalsreduktion/PCA
                X_red = reducer.fit_transform(X_dummy, y_res) # Reduzieren
            else: # Resampler
                X_red, y_res = reducer.fit_resample(X_dummy, y_res) # Resamplen
                y_res = pd.Series(y_res).astype(int)
            X_red = pd.DataFrame(X_red).replace(999999, np.nan) # Dummy zurück zu NaN
            X_imp2 = imputer.fit_transform(X_red) # dann Imputation
            X_imp2, y_res = SMOTE(random_state=42).fit_resample(X_imp2, y_res) # SMOTE anwenden
            y_res = pd.Series(y_res).astype(int) # Zielvariable in Ganzahl (int) umwandeln
            results = evaluate_model(X_imp2, y_res) # CV ausführen
            n_features_out = X_imp2.shape[1] # Anzahl Feautures speichern

        _add_row(f"Kombination: {red_name} → {imp_name}", "Kombination", results, time.time() - start) # Ergebnisse
protokollieren

# 8. Auswertung
gesamt_df = pd.DataFrame(summary_rows) # Liste von Ergebniszeilen in DataFrame umwandeln

# Ergebnistabelle als CSV speichern
csv_path = "Hier Pfad für Speicherort der Ergebnistabelle einfügen" + "experimentierplan_dvv_ergebnisse.csv"
gesamt_df.to_csv(csv_path, index=False) # CSV schreiben (ohne Index)

print(f"\n[Gesamttabelle gespeichert unter] {csv_path}") # Pfad der Ergebnisdatei ausgeben
```

Anhang E: Versionsübersicht Python-Bibliotheken

Tabelle E1: Versionen der verwendeten Python-Bibliotheken

Bibliothek	Version	Funktion im Quellcode
`pandas`	2.3.1	Datenverarbeitung und -strukturierung
`NumPy`	1.25.2	Numerische Berechnungen, Korrelation, Selektion
`scikit-learn`	1.3.2	Imputation, Modellierung, Reduktion, Evaluation
`statsmodels`	0.14.0	Berechnung von VIF-Werten zur Merkmalsauswahl
`imblearn`	0.11.0	SMOTE und ENN